A

Project-I Report

on

DETECTION AND ELIMINATION OF FRAUD RANKING OF MOBILE APPS

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

North Maharashtra University, Jalgaon

Submitted by

Junaid N. Khan
Aalip A. Pinjari
Nourin S. Deshmukh
Nikhil S. Patil

Under the Guidance of

Mr. Dinesh D. Puri

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2016-2017

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project-I entitled *Detection and elimination of fraud ranking of mobile apps*, submitted by

Junaid N. Khan
Aalip A. Pinjari
Nourin S. Deshmukh
Nikhil S. Patil

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of North Maharashtra University, Jalgaon.

Date: October 10, 2016
Place: Jalgaon

Mr. Dinesh D. Puri
Guide

Prof. Dr. Girish K. Patnaik
Head

Prof. Dr. K. S. Wani
Principal

Acknowledgement

Apart from our efforts, the success of any work depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude towards the people who have been instrumental in the successful completion of this project work. We would like to express our sincere gratitude to the Principal Prof. Dr. K. S.Wani, for his encouragement during the work. We would like to express our gratitude to the H.O.D Prof. Dr. Girish K. Patnaik, for his guidance and encouragement during the work. We would like to express gratitude towards our Project Guide Mr. Dinesh D. Puri, for his support and valuable guidance which resulted in the successful completion of project. We would also like to thank Project in charge Mr. Aakash D. Waghmare. We would like to take opportunity to sincerely thanks to all the concerned faculty members, individuals, family members, friends, who made our major project successful. We also thank to all those people who helped us in anyway.

Junaid N. Khan
Aalip A. Pinjari
Nourin S. Deshmukh
Nikhil S. Patil

Contents

List of Figures

Abstract

Now days, mobile apps are ranked by popularity. Rankings get affected by reviews and ratings. As the app get more positive and in favor reviews, its popularity increases. The popularity affects its ranking and attracts more users. Indeed, the reviews manipulation is becoming more common for fraud ranking of mobile apps. The imposter often post fake reviews with a purpose of bumping up the apps in the popularity list. So, the necessity of preventing fraud ranking increases. Such fake reviews must be detected and discarded so that real apps get to their genuine positions in the ranking. For this purpose, the proposed system provide a holistic view of fraud ranking by fake reviews and provide a fake review detection and elimination system for mobile apps. In the experiments, the effectiveness of the proposed system is validated, that shows the scalability of the detection algorithm as well as some regularity of ranking fraud activities.

Chapter 1

Introduction

Introduction chapter introduces the concept, it focusses exactly on what is the meaning of the idea and explains the actual working of it.

Chapter is of seven sections. First Section 1.1 describes background of the project. Motivation is described in Section 1.2.Section 1.3 shows the problem definition. The scope of the project is presented in section 1.4. Section 1.5 describes the various objectives of the project. Overall organization of the project is described in Section 1.6. Section 1.7 gives summary.

1.1 Background

The number of mobile Apps has grown at a panoramic rate over the past few years. as an example, as of the top of april 2013, there are more than 1.6 million Apps at Apple App store and Google Play. To stimulate the development of mobile Apps, several App stores launched daily App leaderboards, that demonstrate the chart rankings of most popular Apps. Indeed, the App leaderboard is one among the most necessary ways for promoting mobile Apps. A higher rank on the leaderboard sometimes results in an enormous number of downloads and million greenbacks in revenue.Therefore, App developers tend to explore various ways in which such as advertising campaigns to market their Apps in order to own their Apps hierarchal as high as possible in such App leaderboards.

However, as a recent trend, rather than counting on tradi-tional marketing solutions, shady App developers resort to some deceitful means to deliberately boost their Apps and eventually manipulate the chart rankings on an App store. this is sometimes enforced by using so-called bot farms or human water armies to inflate the App downloads, ratings and reviews in a very short time. for instance,an article from VentureBeat reportable that, once associate App was promoted with the assistance of ranking manipulation[1][2].

1.2 Motivation

The motivation of the project comes from the fact that fraud ranking in mobile apps is becoming very common today and is a matter of grave concern. Due to which, the financial losses have been increased dramatically.Fraud developers are manipulating the leaderboard by using dishonest means leading to an unfair ranking. The proposed system will detect and eliminate such reviews to provide appropriate ranking chart.

1.3 Problem Defination

A system for detecting and eliminating the fraud ranking of mobile apps is proposed. The apps historical ranking data is given as an input to the system. Then, the leading sessions are identified for each App from its historical ranking records. Then, ranking, rating and reviews based records are extracted from historical data for detecting fraud ranking. Furthur, the sentiment analysis on reviews of the apps is performed. Finally, all the existing evidences are combined for evaluating the credibility of leading sessions from mobile Apps.

1.4 Scope

The scope of the project is limited to the project developers i.e the project is not client centric. This implies that fraud ranking detection and elimination will be entirely done by the systems' management team. The role of the users will be to just post and comment on the reviews which will be basis for further processing by the admin. Also the sentimental analysis module of the project can be used in various other projects in varied fields and hence it is reusable.

1.5 Objective

The major objectives of the project are:

- To detect the fraud reviews
- To eliminate the fake reviews and posts
- To provide a fair ranking leaderboard charts

1.6 Organization Of Report

This section provide the overall layout of the project report on the topic *Detection and elimination of fraud ranking of mobile apps*, means the organization of each and every chapter.

This report contains the Introduction, System Analysis, System Requirements Specification and System Design of the project. *Introduction* chapter no. 1 introduce the concept, it focusses exactly on what is the meaning of the project and explains what is actually the working of the project. All ideas about the project are cleared here. *System Analysis* chapter no. 2 shows overall analysis of the system, description of the system, meaning of the system. In the addition to that literature survey, proposed system and feasibility study, are also described in this chapter. Various system requirements are elaborated in *System Requirement Specifications* chapter no. 3. Study of various system designs like architecture, ER diagrams, DFDs and UML diagrams is also done in the *System Design* chapter no. 4.

1.7 Summary

This chapter shows the introduction of the project. The next chapter provides system analysis of the project.

Chapter 2

System Analysis

System Analysis chapter shows overall system analysis of the concept, description of the system, meaning of the system. Systems analysis is the study of sets of interacting entities, including computer systems analysis. The development of a computer based information system includes a systems analysis phase which produces or enhances the data model which itself is a precursor to creating or enhancing a database. There are a number of different approaches to system analysis. When a computer based information system is developed, systems analysis would constitute the development of a feasibility study, involving determining whether a project is economically, socially, technologically and organizationally feasible.

Chapter is of seven sections. Section 2.1 describes the literature survey of project. Description of system used in project is described in Section 2.2. Section 2.3 describes feasibility study. Risk analysis of the project is presented in Section 2.4. Section 2.5 provides the project scheduling of the project. Effort allocation for the project is described in Section 2.6 and finally Section 2.7 gives summary.

2.1 Literature Survey

In the literature, some related work in this area is studied, such as web ranking spam detection, online review spam detection and mobile App recommendation. These are discussed in brief in Subsection 2.1.1, Subsection 2.1.2 & Subsection 2.1.3 respectively.

2.1.1 Web Ranking Spam Detection

Web ranking spam refers to any deliberate actions which bring to selected webpages an unjustifiable favorable relevance or importance. For example, Ntoulas et al. have studied various aspects of content-based spam on the web and presented a number of heuristic methods for detecting content based spam. Zhou et al. have studied the problem of unsupervised web ranking spam detection. Specifically, they proposed an efficient online link spam and

term spam detection methods using spamicity. Recently, Spirin and Han have reported a survey on web spam detection, which comprehensively introduces the principles and algorithms in the literature. Indeed, the work of web ranking spam detection is mainly based on the analysis of ranking principles of search engines, such as PageRank and query term frequency. This is different from ranking fraud detection for mobile Apps[3][4].

2.1.2 Online Review Spam Detection

This category focusses on detecting online review spam. For example, Lim et al. have identified several representative behaviors of review spammers and model these behaviors to detect the spammers. Wu et al. have studied the problem of detecting hybrid shilling attacks on rating data. The proposed approach is based on the semisupervised learning and can be used for trustworthy product recommendation. Xie et al. have studied the problem of singleton review spam detection. Specifically, they solved this problem by detecting the co-anomaly patterns in multiple review based time series. Although some of above approaches can be used for anomaly detection from historical rating and review records, they are not able to extract fraud evidences for a given time period (i.e., leading session)[5][6]..

2.1.3 Mobile App Recommendation

This category includes the study on mobile App recommendation. For example, Yan and Chen developed a mobile App recommender system, named Appjoy, which is based on users App usage records to build a preference matrix instead of using explicit user ratings. Also, to solve the sparsity problem of App usage records, Shi and Ali studied several recommendation models and proposed a content based collaborative filtering model, named Eigenapp, for recommending Apps in their website Getjar. In addition, some researchers studied the problem of exploiting enriched contextual information for mobile App recommendation. For example, Zhu et al. proposed a uniform framework for personalized context-aware recommendation, which can integrate both context independency and dependency assumptions. However, to the best of our knowledge, none of previous works has studied the problem of ranking fraud detection for mobile Apps[7][8]..

2.2 Proposed System

As the number of mobile Apps are increasing at a breathtaking rate over the past few years. So the competition and comparison between them is also growing. To stimulate the development of mobile Apps, many App stores launched daily App leaderboards, which demonstrate the chart rankings of most popular Apps. A higher rank on the leaderboard

usually leads to a huge number of downloads and million dollars in revenue. Therefore, App developers performing various fraudulent activities and try to boost their apps top in such App leaderboards. This is called fraud ranking.

The proposed system is a solution to this problem by detecting the fraud ranking. In this system, the download information is an important signature for detecting ranking fraud. Therefore, evidences from Apps historical ranking, rating and review records for ranking fraud detection are extracted. Then, it detect ranking fraud happened in Apps historical leading sessions. Finally, the linear combination of all the evidences extracted is performed.

2.3 Feasibility Study

The feasibility study is an evaluation and analysis of the potential of a proposed project which is based on extensive investigation and research to support the process of decision making.Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success.

2.3.1 Economical Feasibility

Feasibility studies are crucial during the early development of any project and form a vital component in the business development process. Accounting and Advisory feasibility studies enable organizations to assess the viability, cost and benefits of projects before financial resources are allocated. They also provide independent project assessment and enhance project credibility. In current research, it is common to use a ranking algorithm to model how once a country reaches a specificc level of economic development, the configuration of structural factors, such as size of the commercial bourgeoisie, the ratio of urban to rural residence, the rate of political mobilization, etc., will generate a higher probability of transitioning from authoritarian to democratic regime.

2.3.2 Operational Feasibility

In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. The objective of the proposed approach is to utilize the strengths of semantic analysis Model to complement the weaknesses of other classification techniques. Consequently, instead of finding single isolated patterns, we focus on understanding the relationships between these patterns. The proposed approach was evaluated with a case study. Experimental evaluation proves the feasibility and effectiveness of the approach.

2.3.3 Technical Feasibility

The purpose of the study is to provide the necessary information to enable the company (and Enterprise) to come to conclusions regarding the project's viability. We show that detection and prevention system can be adapted to handle fraud ranking of mobile apps. This helps in achieving scalability. We represent this using semantic analysis. This provides significant improvement in the performance.

2.4 Risk Analysis

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem it might happen, it might not. But, regardless of the outcome, its a really good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur. Everyone involved in the software process managers, software engineers, and customers participate in risk analysis and management. Before embarking on the project it is necessary to review all of the risks that might be involved in it. These risks have been documented before the coding of the project started.

The majority of the risks lie under the categories:

2.4.1 Software Risks

Although there has been considerable debate about the proper definition for software risk, there is general agreement that risk always involves two characteristics.

- Uncertainty-the risk may or may not happen; that is, there are no 100%
- Loss-if the risk becomes a reality, unwanted consequences or losses will occur

When risks are analyzed, it is important to quantify the level of uncertainty and the degree of loss associated with each risk. To accomplish this, different categories of risks are considered.

2.4.2 Project Risks

Project risks threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirements problems and their impact on a software project.

In the project, project risk occurs if our requirement of technical member means technical team is unavailable according to our project plan and estimation and if our project is not completed within time, in this situation project risk can occur.

2.4.3 Technical Risks

Technical risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification, and maintenance problems. In addition, specification ambiguity, technical uncertainty, technical obsolescence, and "leading-edge" technology are also risk factors. Technical risks occur because the problem is harder to solve than we thought it would be. In our project, if any module is not working properly or module to module linking is not according to the expectations, then technical risk may occur.

- List of technical risk in the project:

  Network failure: The proposed system requires the network connections for its working. If some failure occurs in the working of network, then the network may fail and it affect working of the system. This problem of network failure can be solved by using proper network maintenance & alternate network availability.

2.5 Project Scheduling

Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. It is important to note, however, that the schedule evolves over time. During early stages of project planning, a macroscopic schedule is developed. The schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into a detailed schedule.

Here, specific software tasks (required to accomplish an activity) are identified and scheduled. Scheduling for software engineering projects can be viewed from two rather different perspectives. In the first, an end-date for release of a computer-based system has already (and irrevocably) been established. The software organization is constrained to distribute effort within the prescribed time frame. The second view of software scheduling assumes that rough chronological bounds have been discussed but that the end-date is set by the software engineering organization. Effort is distributed to make best use of resources and an end-date is defined after careful analysis of the software. Unfortunately, the first situation

is encountered far more frequently than the second. The Gantt Chart for project scheduling is shown in Figure 2.1

Task Name		July				August				September				October
		W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1
Requirement Gathering	S	✓	✓	✓										
	C	✓	✓	✓										
Analysis of Project	S			✓	✓	✓	✓							
	C			✓	✓	✓	✓							
Design of Project	S						✓	✓	✓	✓	✓			
	C						✓	✓	✓	✓	✓			
Documentation Of Project	S										✓	✓	✓	✓
	C										✓	✓	✓	✓

S: Schedule, C: Completion

Figure 2.1: Gantt Chart for Project Scheduling

2.6 Effort Allocation

Project means team work; Project is developed by combination of effort of team. So whole project is divided into modules and number of modules is allotted to team members. After completion of each module, it is link from one module to another module to form a complete project. The effort allocation should be used as a guideline only. The characteristics of each project must dictate the distribution of effort. Work expended on project planning rarely accounts for more than 23 percent of effort, unless the plan commits an organization to large expenditures with high risk. Requirements analysis may comprise1025 percent of project effort. Effort expended on analysis or prototyping should increase in direct proportion with project size and complexity. A range of 20 to 25percent of effort is normally applied to software design. Time expended for design review and subsequent iteration must also be considered. The Effort Allocation Table for the proposed system is shown in the following

Figure 2.2

Phases	Activity	Junaid N. Khan	Aalip A. Pinjari	Nourin S. Deshmukh	Nikhil S. Patil
Requirement Gathering	Identification of Project and Requirement Gathering	✓	✓	✓	✓
	Study of Existing System	✓	✓	✓	✓
	Study of Process Model and Effort Allocation	✓		✓	
Analysis	Identification of Functional and Non-functional Requirements		✓		✓
	Data Modeling		✓	✓	
	Functional Modeling	✓			✓
	Behavioral Modeling			✓	✓
Design	Data Design	✓	✓		
	Architecture Design	✓		✓	
	Interface Design		✓	✓	
	Component Level Design	✓			✓

Figure 2.2: Effort Allocation Table

2.7 Summary

This chapter shows the system analysis of the project. The next chapter provides various system requirement specifications.

Chapter 3

System Requirement Specification

System Requirement Specifications chapter provides various requirements of the project such as functional,nonfunctional,software and hardware requirements. Understanding user requirements is an integral part of information systems design and is critical to the success of interactive systems. It is now widely understood that successful systems and products begin with an understanding of the needs and requirements of the users. User-centered design begins with a thorough understanding of the needs and requirements of the users. The benefits can include increased productivity, enhanced quality of work, reductions in support and training costs, and improved user satisfaction.

Chapter is of seven sections. Section 3.1 describes the required hardware for project. Required software for project is described in Section 3.2. Section 3.3 shows the functional requirements. The Non-Functional Requirements of project are presented in Section 3.4. Section 3.5 provides other requirements and various constraints and finally Section 3.6 gives summary.

3.1 Hardware Requirements

1. Processor - Intel(R)Core(TM)2 Duo
2. Hard disk - 40 GB
3. RAM - 2GB
4. Monitor - 15 VGA color
5. Mouse - Logitech

3.2 Software Requirements

1. Front End - Java 1.7.[9]

2. Back End - My SQL,[10]

3. Platform - Win XP/7, Linux, Netbeans, Text Pad.

3.3 Functional Requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

The functional requirements of the proposed system include the apps historical data which is the combination of rankings, reviews and ratings data. The the sufficient number of apps must be available. proper network connection must available without any interrupt.

3.4 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing

The Nonfunctional requirements of the proposed system includes those functions which does not effect on function and behavior of project for desired goal and objective of project. Non- functional requirement just provides user friendliness and notifications that are not most necessary for the project.

3.5 Other Requirements and Constraints

Requirements analysis is not a simple process. Particular problems faced by the analyst are:

1. Addressing complex organizational situations with many users.

2. Users and designers thinking along traditional lines, reacting the current system and processes, rather than being innovative

3. Users not knowing in advance what they want from the future system.

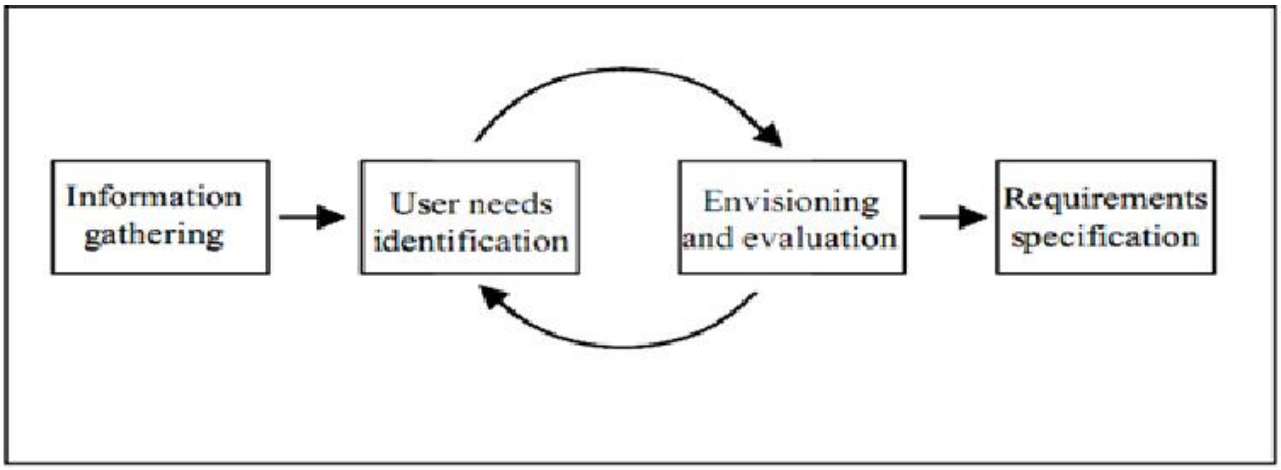

Figure 3.1: General Process for User Requirement Analysis

This section considers how these problems can be addressed by selecting appropriate methods to support the process of user requirements generation and validation. It describes each method briefly and shows how it contributes to the requirements process. The basis for the application of different user requirements methods is a simple process as shown in figure3.1;

3.6 Summary

This chapter shows the various system requirement specifications of the project. The next chapter provides system design of the project.

Chapter 4

System Design

System Design chapter provides graphical structure of the project by using various UML diagrams. System design provides the understanding and procedural details necessary for implementing the system recommended in the system study. Design is a meaningful engineering representation of something that is to be built. It can be traced to a customers requirements and at the same time assessed for quality against a set of predefined criteria for good design. In the software engineering context, design focuses on four major areas of concern are data, architecture, interfaces and components.

Chapter is of six sections. Section 4.1 describes system architecture of the project. E-R Diagram is described in Section 4.2. Section 4.3 describes the database design. The data flow diagram of the project is presented in Section 4.4. Section 4.6 shows various UML diagrams, and finally Section 4.7 gives summary.

4.1 System Architecture

System architecture alludes to the overall structure of the software and the ways in which that structure provides conceptual integrity for a system. In its simplest form, architecture is the hierarchical structure of program components (modules), the manner in which these components interact and the structure of data that are used by the components. The architectural design representation defines the components of a system (e.g., modules, objects) and the manner in which those components are packaged and interact with one another. The architecture of the proposed system is shown in Figure 4.1.

It include the following module:

1. Input : The input is the apps historical data
2. Leading Session : It identifies the leading events when the fraud ranking happen
3. Ranking : Ranking related historical data to detect fraud

4. Review : Review Manipulation data to detect fraud

5. Rating : Rating related historical data to detect fraud

6. Aggregation : Linear combination of all existing evidances

The apps historical data is given to the system as a input, from which leading events are discovered i.e. when the fraud is happen. Then various evidences are extracted from the data, such as ranking, reviews & rating related. All these evidences are combined and fraud is detected.

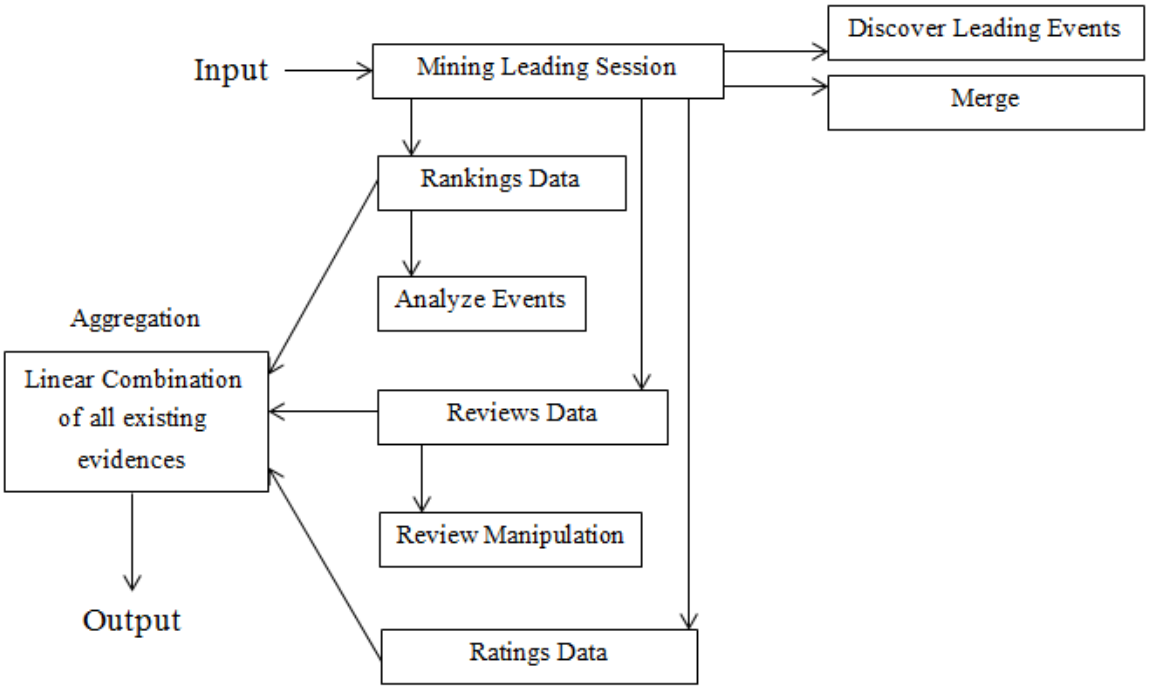

Figure 4.1: System Architecture

4.2 E-R Diagram

The entity relationship data model is based on a perception of a real world that consist of a collection of basic objects called entities, and relation among these objects.

Figure 4.2 shows the E-R Diagram for the Proposed System.

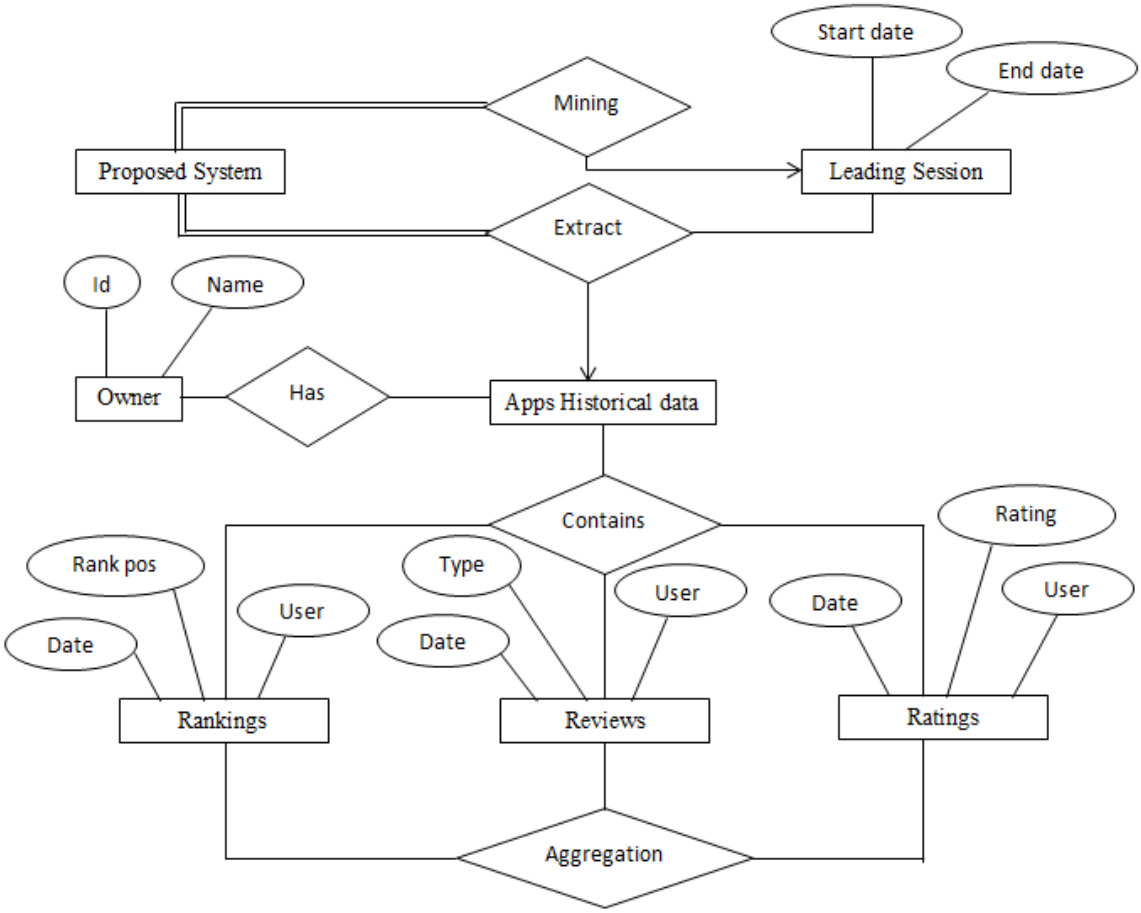

Figure 4.2: E-R Diagram

4.3 Database Design

The database design contains the database schema of the project.

4.3.1 Schema

Database schema shows the various tables used in database. Database schema is as shown in figure 4.3.

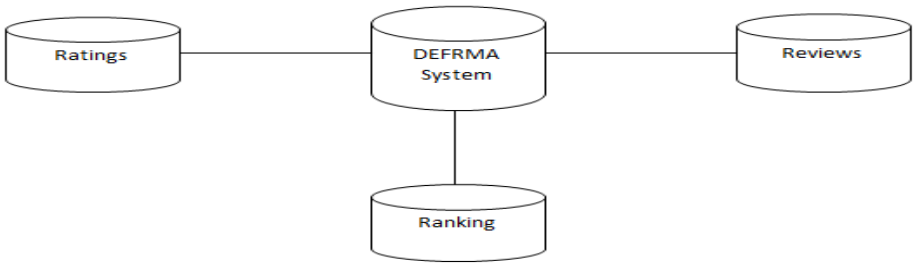

Figure 4.3: Database Schema

4.4 Data Flow Diagram

The DFD is an input process output view of a system i.e. data objects into the software, are transformed by processing elements, and resultant data objects out of the software. The DFD enables the software engineer to develop models of the information domain and functional domain at the same time. As the DFD is refined into greater levels of details, the analyst perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

The level 0 DFD of the system is shown in Figure 4.4.

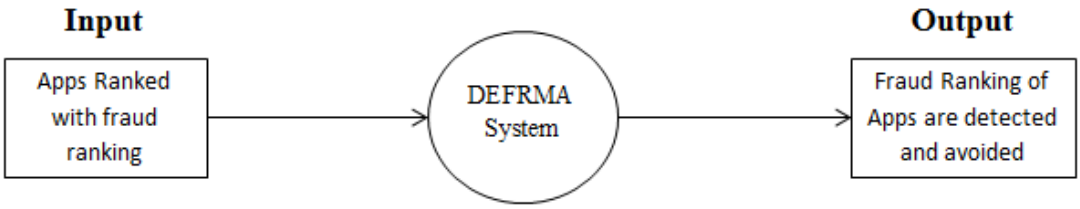

Figure 4.4: Data Flow Diagram(DFD) level 0

Figure 4.4 shows the level 1 DFD of the system.

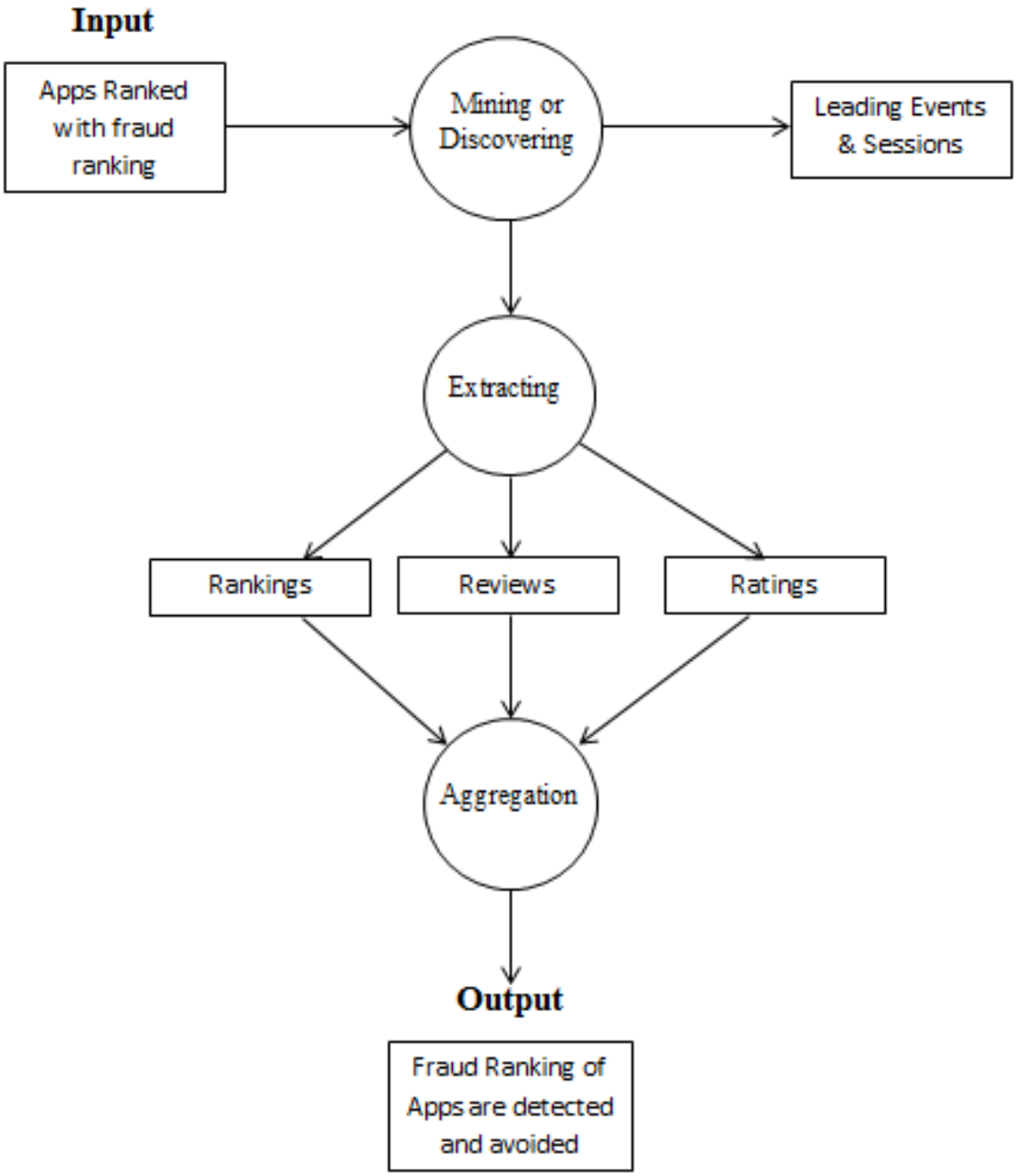

Figure 4.5: Data Flow Diagram(DFD) level 1

And finally Level 2 DFD of the system is shown in Figure 4.6

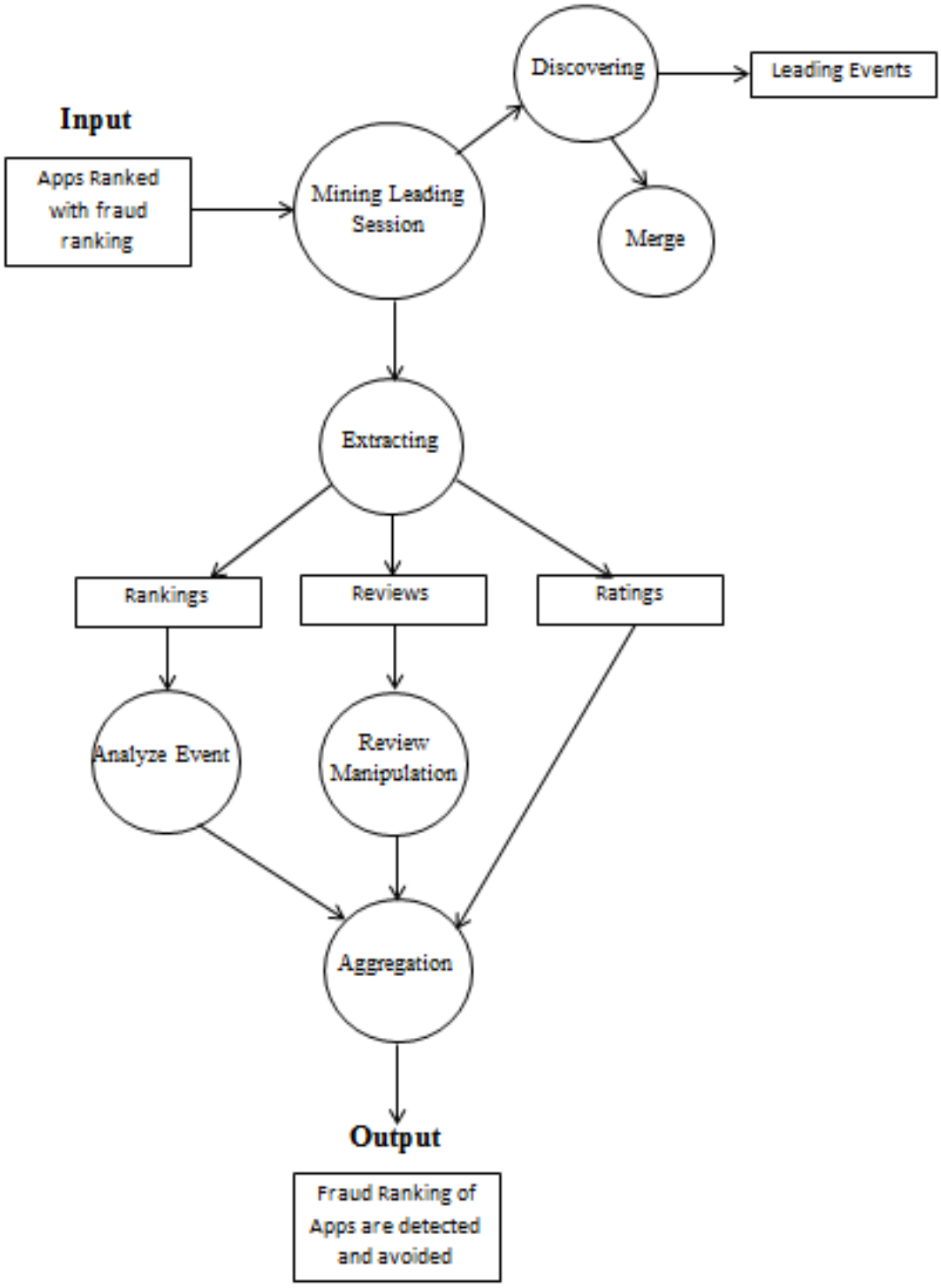

Figure 4.6: Data Flow Diagram(DFD) level 2

4.5 Interface Design

The interface design describes how the software communicates within itself, with systems that inter operate with it, and with humans who use it.Interface design focuses on three

areas of concern:

- The design of interfaces between software components
- The design of interfaces between the software and other non human producers and consumers of information (i.e., other external entities)
- The design of the interface between a human (i.e., the user) and the computer

4.5.1 Introduction to Interface Design

User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

4.5.2 Component of Interface Design

A software engineer designs the user interface by applying an iterative process that draws on prede

ned design principles.

4.5.3 Need of Interface Design

If software is difficult to use, if it forces you into mistakes, or if it frustrates your efforts to accomplish your goals, you wont like it, regardless of the computational power it exhibits or the functionality it offers. Because it molds a users perception of the software, the interface has to be right.

4.6 UML Diagrams

The Unified Modeling Language is a language that defines the industry's best engineering practices for the modeling systems. The goal of UML is to be a ready-to-use expressive visual modeling language that is simple and extensible.

4.6.1 Use Case Diagram

Use case diagram shows a set of use cases, actors and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviour of the system. figure4.7 shows use case diagram of the project.

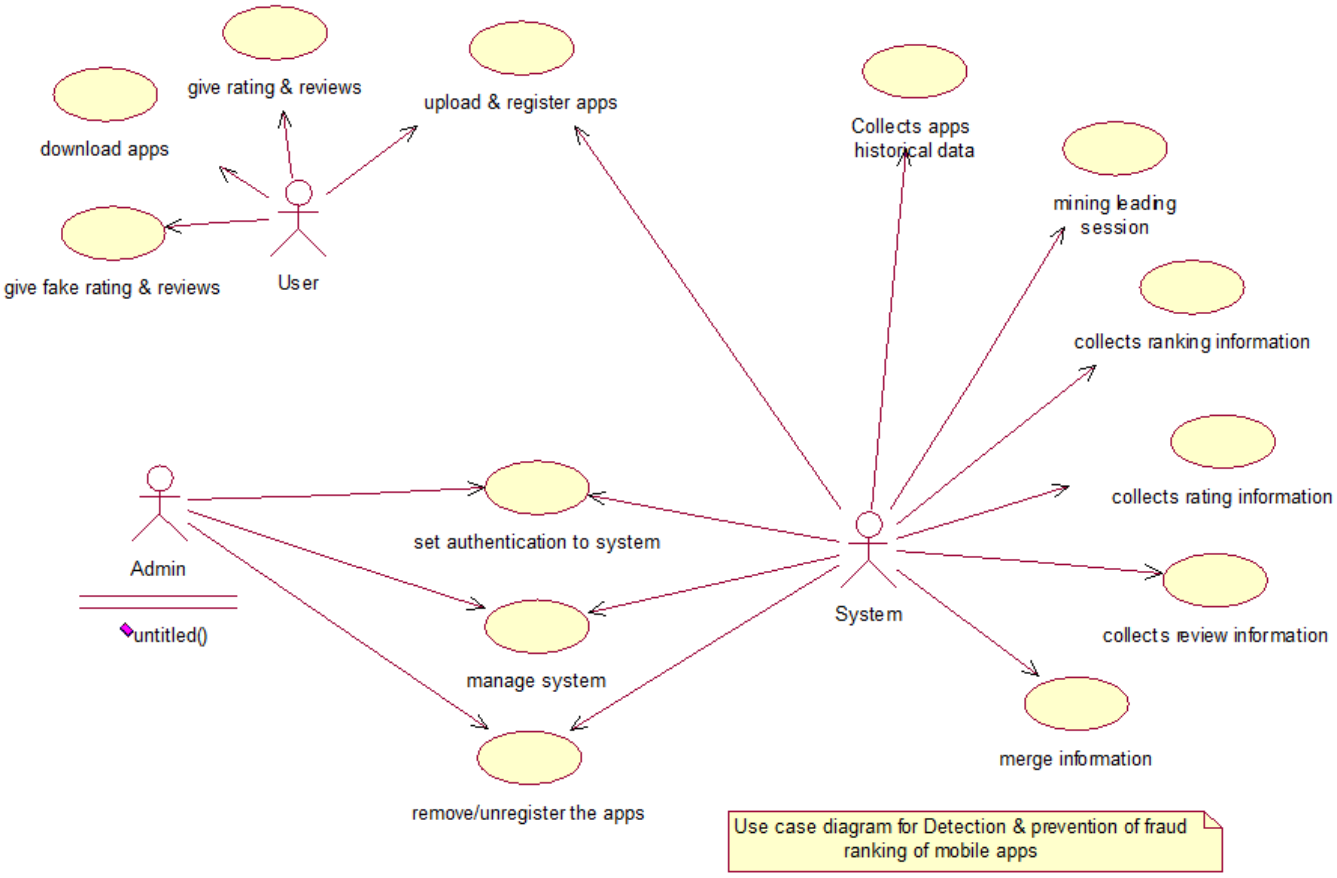

Figure 4.7: Use Case Diagram

4.6.2 Class Diagram

A class diagram shows a set of classes, interfaces, collaborations and their relationships. Class diagram address the static design view of a system. Class diagrams are important not only for visualizing, specifying and documenting structural models but also for constructing executable systems. Figure4.8 shows class diagram of the project.

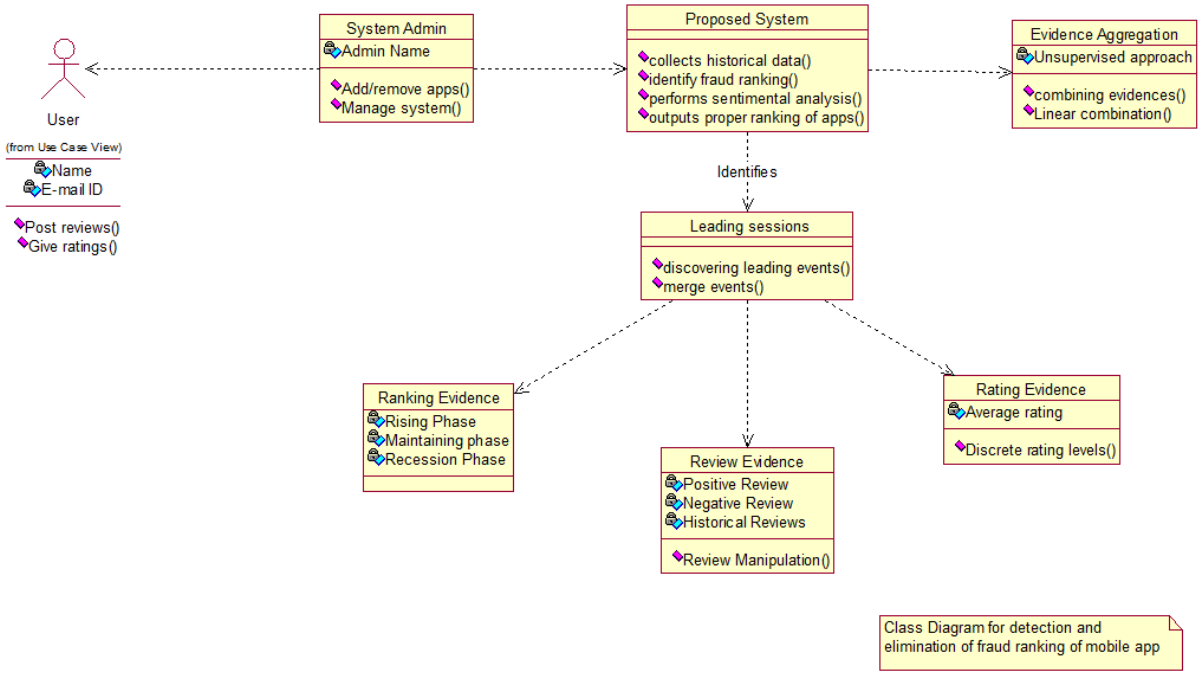

Figure 4.8: Class Diagram

4.6.3 Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Sequence diagram is isomorphic means that we can take one and transform it into the other. Figure4.9 shows the sequence diagram of the project.

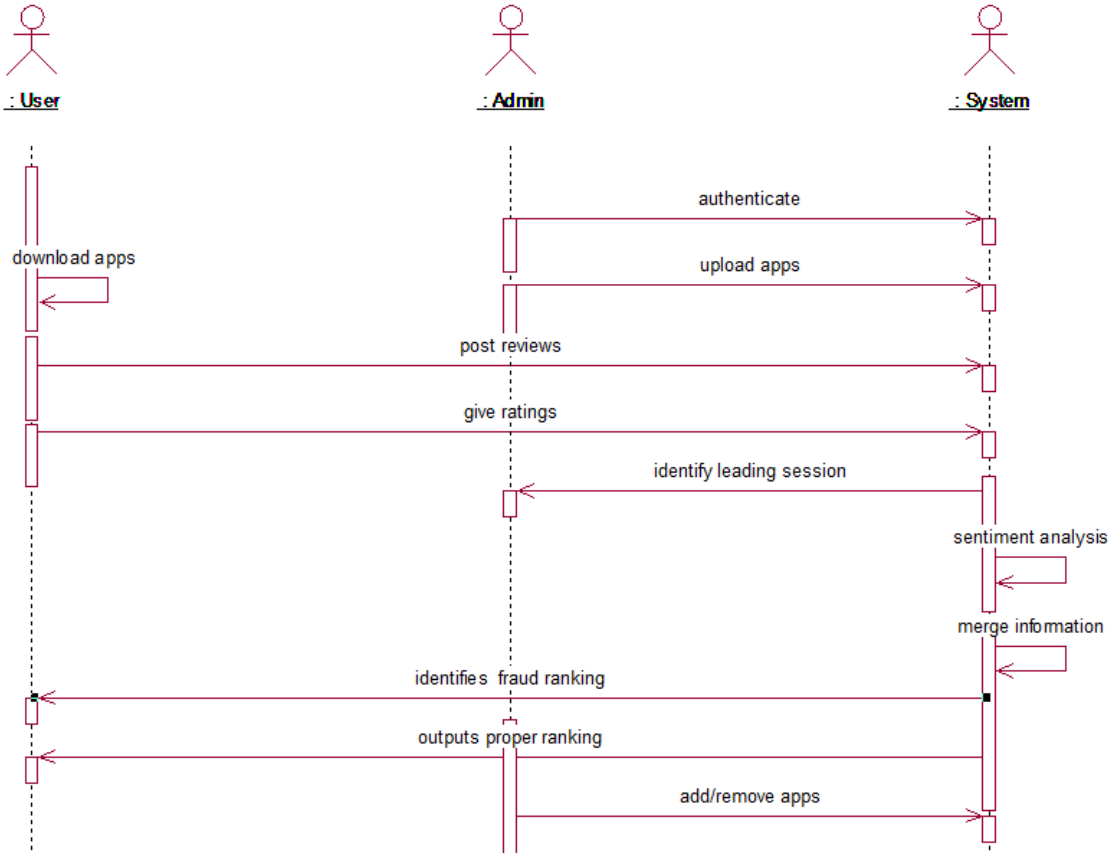

Figure 4.9: Sequence Diagram

4.6.4 Collaboration Diagram

Figure4.10 shows the collaboration diagram of the project.

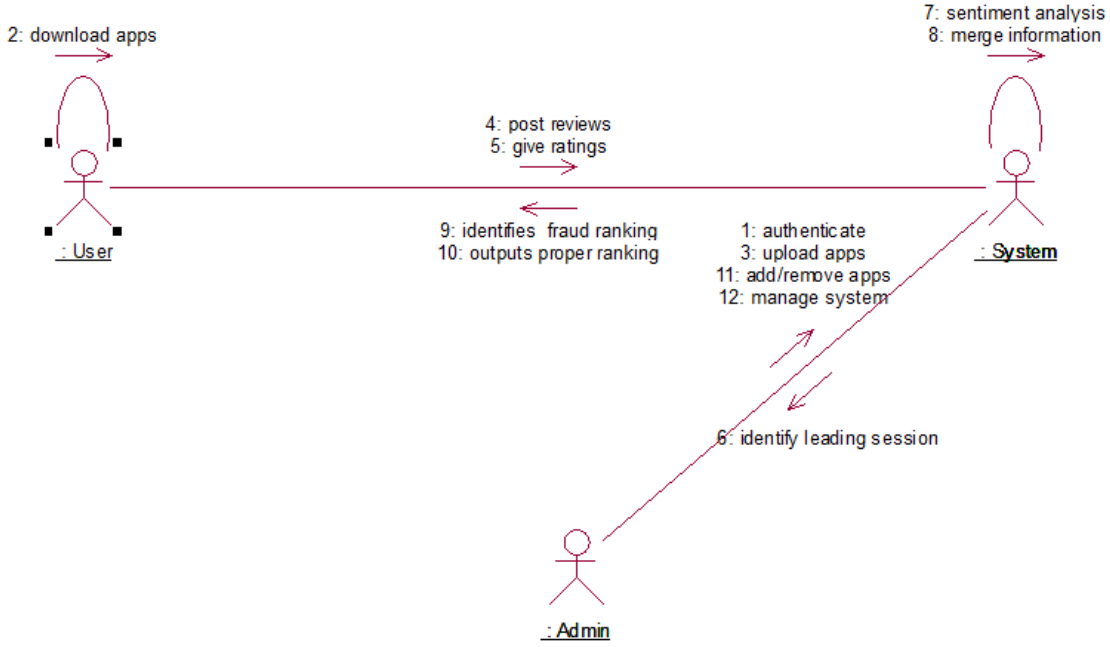

Figure 4.10: Collaboration Diagram

4.6.5 Component Diagram

A component diagram shows the organizations and dependencies among a set of components. Component diagram address the static implementation view of a system. They are related to class diagram in that a component typically maps to one or more classes, interfaces or collaborations. Figure4.11 shows component diagram of the project.

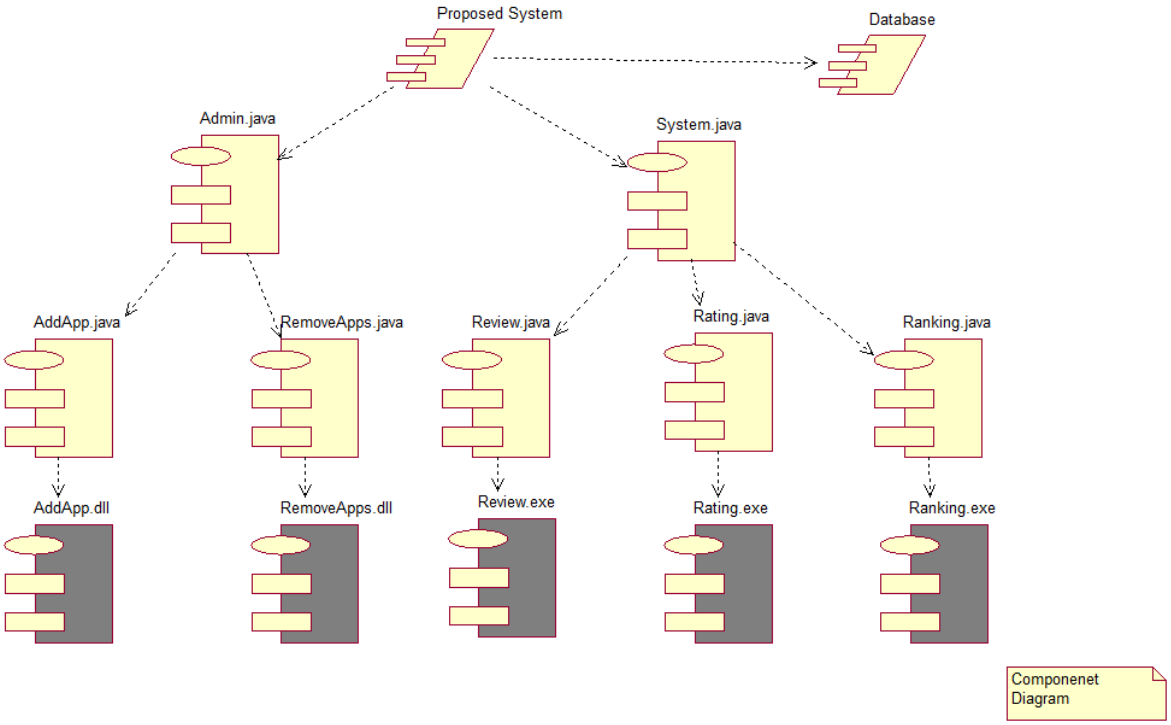

Figure 4.11: Component Diagram

4.7 Summary

This chapter shows the system design of the project.

Conclusion

The Detection and Elimination of fraud ranking of mobile apps project is designed to provide genuine rankings of mobile apps. Generally it is observed that fraud ranking does not occur always but only in some leading session. So, the leading sessions are identified for each App from its historical ranking records. Then, ranking, rating and reviews based records are extracted from historical data for detecting fraud ranking. Finally, all the existing evidences are combined for evaluating the credibility of leading sessions from mobile Apps. This, in turn, will improve user experience of using mobile apps. Fraud ranking is eliminated leading to a fair app-leaderboard. The effectiveness of this project can be proved from the experimental results that will be obtained. The project can be extended with other mobile app services such as mobile app recommendation for enhancing user experience.

Bibliography

[1] Hengshu Zhu, Hui Xiong, Yong Ge and Enhong Chen, Discovery of Ranking Fraud for Mobile Apps, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 1, JANUARY 2015, Accessed on: 30 July, 2016.

[2] Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong and Enhong Chen, Popularity Modeling for Mobile Apps: A Sequential Approach, Popularity Modeling for Mobile Apps: A Sequential Approach, Accessed on: 5, August 2016.

[3] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, Detecting spam web pages through content analysis, in Proc. 15th Int. Conf. World Wide Web, 2006, pp. 8392, Accessed on: 12, August 2016.

[4] N. Spirin and J. Han, Survey on web spam detection: Principles and algorithms, SIGKDD Explor. Newslett., vol. 13, no. 2, pp. 5064, May 2012, Accessed on: 12, August 2016.

[5] N. Spirin and J. Han, Survey on web spam detection: Principles and algorithms, SIGKDD Explor. Newslett., vol. 13, no. 2, pp. 5064, May 2012, Accesed on: 18, August 2016.

[6] N. Spirin and J. Han, Survey on web spam detection: Principles and algorithms, SIGKDD Explor. Newslett., vol. 13, no. 2, pp. 5064, May 2012, Accessed on: 18, August 2016.

[7] N. Spirin and J. Han, Survey on web spam detection: Principles and algorithms, SIGKDD Explor. Newslett., vol. 13, no. 2, pp. 5064, May 2012, Accessed on: 28, August 2016.

[8] N. Spirin and J. Han, Survey on web spam detection: Principles and algorithms, SIGKDD Explor. Newslett., vol.13, no. 2, pp. 5064, May 2012, Accessed on: 28, August 2016.

[9] Java, "http://www.java.com/", Accessed on: 10, September 2016.

[10] MySQL, "http://www.mysql.com", Accessed on: 17, September 2016.