

Software Recommendation System

Sanjana Deore, Komal Mahale, Salwa Shaikh, Zarlina Sayyed

Department Of Computer Engineering, Guru Gobind Singh Polytechnic Nashik

● Abstract

System software is software designed to provide a platform for other software. Examples of system software include operating systems like macOS, GNU/Linux, Android and Microsoft Windows, computational science software, game engines, industrial automation, and software as a service application.

In contrast to system software, software that allows users to do user-oriented tasks such as create text documents, play games, listen to music, or browse the web are collectively referred to as application software.

Software developers must interact with large amounts of different types of information and perform many different activities to build a software system. To ease the finding of information and hone workflows, there has been growing interest in building recommenders that are intended to help software developers work more effectively. Building an effective recommender requires a deep understanding of the problem that is the target of a recommender, analysis of different aspects of the approach taken to perform the recommendations and design and

evaluation of the mechanisms used to present recommendations to a developer.

Keywords: macOS, software, programming, OS, web browser.

● Introduction

In the software recommendation system, we Provide Software by Category wise. This system recommends essential software according to your requirement in the Software System. You can download the software under the software category. Help users to find software, or services such as Firefox, chrome, Microsoft office provide a platform for other software.

System software is software designed to provide a platform for other software. Examples of system software include operating systems like MacOS, GNU/Linux, Android and Microsoft Windows, computational science software, game engines, industrial automation, and software as a service application.

In most applications, software was custom-written by computer users to fit their specific hardware and requirements. In contrast, system software was usually supplied by the manufacturer of the computer hardware.

- **Literature Survey**

Recommendation of software systems have become an important research field since recommender systems are defined as the supporting systems which help users to find software, or services (such as Firefox, chrome, Microsoft office). The provision of personalized recommendations however requires that the system knows something about every user. Every recommender system must develop and maintain a user model or user profile.

A recommendation system for software is software application that provides software. We review the key decisions in software recommender systems: the user downloads the software. The ways in which prediction quality is measured, the evaluation of prediction attributes other than quality, and the user-based evaluation of the system as a whole. A set of dimensions that distinguish, describe and categorize multi-criteria software recommender systems.

- **Existing System**

In the existing system currently there is no online web application available to report crime online. In order to report any complaints related to crime, people have to contact the nearest police station. People of the particular city are not aware of crime related things such as list of Most Wanted criminals of their city, latest crime related news, missing persons of their area etc... People have to view News Channels or Read News Paper for such crime related information. Thus we can say that existing system is manual and does not provide all the information from one source In the existing system users can download the software but this system is not

providing category and subcategory view software and does not provide steps on how to download the software steps for the user.

There has been a lot of work done in this field. For example, one very popular algorithm is Collaborative Filtering. One type of collaborative filtering is user-based collaborative filtering, which starts by finding a set of customers who have purchased and rated similar items with the target user's purchasing history.

- **Problem Definition**

In Software Recommendation System, you can download software by Selecting its Category and then subcategory. After selecting both category and subcategory users can download it. A recommender system or a recommendation engine is a subclass of an information filtering system that tries to make predictions on user preferences and also make recommendations which should interest customers.

In this technology-driven era, the use of recommendation is significantly wide. This project is based on the database, web based techniques. To keep the records in the database it uses MySQL software which is one of the best and the easiest databases to keep information.

This project uses Bootstrap as the front-end Web application which is an PHP based Programming and has connectivity with MySQL. The user's and Software details are stored in a database like their name, date of birth and gender and Software name, Software Link.

1. Provide Software:
2. Manage User Details:
3. Manage Category and Subcategory Details:

4. Manage Software Details:
5. Admin Module:

- **System Architecture**

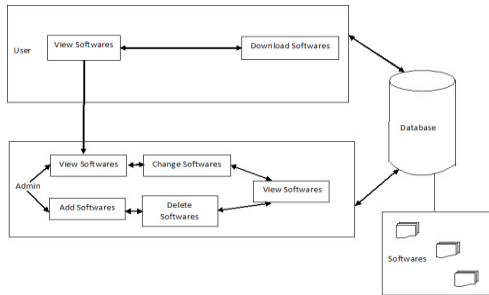


Fig-1: System Architecture Diagram

- **System Requirement**

Hardware Used:

1. Processor : Intel core i3
2. Ram: 4 GB (min)
3. Hard disk : 128GB

Software Used

1. Operating System: Window 7.
2. Front End: Bootstrap4,css,Html
3. Back End: MySQL, PHP
4. Code editor: Visual Studio, Sublime text

- **Advantages**

1. Reduced time to benefit. Software as a service (SaaS) differs from the traditional model because the software (application) is already installed and configured.
2. Scalability and integration
3. New releases (upgrades)
4. Easy to use and perform proof-of-concepts.
5. Better customer service

- **Disadvantages**

1. It requires an active internet connection else error may occur.

- **Application:**

1. Embedded and Real-Time Systems.
2. One of the application areas of software where correctness is more critical is embedded systems
3. Software finds its presence in enterprise as well as non-enterprise applications

- **Future Scope:**

- As we can see in previous research there is lots of improvement is needed so there is lots of future work are there where research can work on it and improve they result.
- To develop a system which reduce the issue of search software by their categories
- Software Recommendation system has been an active area of research for a decade or so and continues to be a software's.

- **Conclusion**

A good software reuse process facilitates the increase of productivity quality and reliability and decreases the costs and implementation time by far the most important part of the reuse process is the people if the people in the organization do not understand the concepts behind reuse and do not see the benefits, reuse won't happen reuse processes and procedures must be incorporated into the existing software development process. We have tried our best to improve

recommendations for “cold users” but still recommendations given to them largely depend on the responses provided by the user to us through a form but it can be improved by considering the region, ethnicity, and neighbour users. We are looking for algorithms being developed and as soon as any algorithm comes, we will try to implement that algorithm in our project. Every user expects a very simple but elegant UI design, good recommendations so that he has good options to choose from, better search results and accurate product details.

● **References:**

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record. vol. 22, pp. 207–216. ACM (1993)
2. Amman, S., Porsche, S., Mazzini, M.: Method-call recommendations from implicit developer feedback. In: Proceedings of the 1st International Workshop on Crowd-Sourcing in Software Engineering. pp. 5–6. ACM (2014)
3. Amusable, D.P.: The psychology of meaningful verbal learning. (1963)
4. Amusable, D.P., Novak, J.D., Hansen, H., et al.: Educational psychology: Cognitive view (1968)
5. Berg, E.A.: A simple objective technique for measuring flexibility in thinking. The Journal of general psychology 39(1), 15–22 (1948).
6. Beyer, H., Holtzblatt, K.: Contextual design: A customer-centered approach to systems designs (Morgan Kaufmann series in interactive technologies) (1997)
7. Felsenthal, D. S., & Mach over, M. (1998). The measurement of voting power. Books.
8. Booth, W.C., Colombo, G.G., Williams, J.M.: The craft of research. University of Chicago press (2003)
9. Bruch, M., Mongers, M., Mazzini, M.: Learning from examples to improve code completion systems. In: Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. pp. 213–222.

Web Reference:

1. <https://www.w3schools.com/>
2. <https://www.geeksforgeeks.org/>