Advances in Computer, Signals and Systems (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2024.080303
ISSN 2371-8838 Vol. 8 Num. 3

Design of personalized action recommendation system based on mobile platform

Chen Yuqi, Yang Zhengshuai, Yan Haiwei*

School of Computer and Information Engineering, Nantong Institute of Technology, Nantong, 226001, China
**Corresponding author*

Keywords: Personalized actions, Recommendation system, Mobile platform, Time period

Abstract: This paper describes the design of a personalized action recommendation system based on mobile platform. With the development of the digital era, the application of mobile smart devices is becoming more and more widespread, and people's demand for mobile applications continues to grow. The system designed in this paper aims to provide personalized action recommendations based on time periods and users' historical preferences to improve users' activity planning and usage experience. The system uses Xamarin as the development framework, C# as the development language and MySQL as the database. The main functions include dynamic recommendation, action categorization, search function, and extraction of action recommendations. By analyzing the user's historical data and combining the characteristics of the time period, the system is able to generate personalized recommendations and improve the accuracy and efficiency of recommendations. In addition, the system also includes an administrator module for supervising and managing dynamic content to ensure the normal operation of the system.

1. Introduction

With the development of the digital era, the development of Internet technology has been accelerating, more and more mobile smart devices are applied to people's daily production and life[1] , and people also rely more and more on mobile terminals and applications to complete various tasks and activities. At the same time, users' demand for mobile applications is also growing. In this context, personalized recommendation system based on mobile platform becomes an important tool to satisfy users' needs. In this paper, we design a personalized action recommendation system based on time periods and users' historical preferences to improve users' activity planning and usage experience.

Through this system, users can get the best action recommendation suitable for the current time period according to their schedule and personal preference. At the same time, the time period-based action recommendation system also provides operators with marketing and promotion opportunities to attract consumers by releasing special offers for different time periods, thus promoting business development and market competition.

In traditional recommender systems, the user's interests and preferences[2] are mainly considered, while the influence of the time factor is neglected. In contrast, this system combines the characteristics of time periods for temporal recommendations, thus providing more accurate recommendations. By

providing recommendations for actions that are most suitable for the user's current time period, it helps the user to better schedule activities, which helps the user to effectively plan their time and optimize the use of resources.

The main objective of the article is to elucidate the development and design of a personalized action recommendation system. This system helps the user to get the best action recommendation for the current time period.

2. Key technologies for personalized action recommendation systems

During the design of this system, Xamarin was chosen as the development framework, C# as the development language, and MySQL as the database.

2.1 Xamarin cross-platform framework

Xamarin is a cross-platform framework for mobile platform development based on the C# language and the . Developers can also easily access device hardware features and APIs through the Xamarin. Meanwhile, Xamarin provides the Xamarin.Forms toolkit, which can be used to simplify cross-platform application development. By choosing Xamarin.Forms, developers only need to write a set of user interface code, which can run on multiple platforms, greatly reducing the development workload and maintenance costs, and significantly improving the speed of application development. [3-4] .

2.2 C# Development Language

C# is a general-purpose, object-oriented programming language introduced by Microsoft, which runs on the .NET platform and is widely used in the development of various applications. C# has the advantages of strong type-checking and rich language features, and can be used to develop desktop applications, Web applications, mobile applications and other application scenarios.

By using the C# language and Xamarin. Forms, developers can write a shared set of user interface and application logic code, allowing programs to run on multiple mobile platforms. At the same time, the strongly typed nature of C# matches the data structures of the MySQL database, providing better type safety and code organization.

2.3 MySQL Database

MySQL is an open source relational database management system (RDBMS) with good performance, reliability and scalability [5], capable of handling large amounts of data and concurrent requests, and is widely used for database management in Internet applications. MySQL databases are used as the back-end database of choice to provide persistence and management capabilities for Xamarin applications. Using the MySQL connector to connect to a MySQL database, database operations such as query, insert, update, and delete can be performed, which is valuable for building the data access layer in a recommender system.

In addition, C# and MySQL can be seamlessly integrated with the Xamarin framework during development using the Xamarin framework, which allows developers to develop cross-platform mobile applications with a unified set of technology stacks with reliable data storage and efficient access.

3. Requirements analysis for the design of personalized action recommendation system

3.1 Needs analysis

(1) Time segmentation

In order to meet the different needs of users in different time periods, the system needs to divide the user's action time period (morning, morning, afternoon, evening) according to different time periods, the specific time division is shown in Table 1.

Table 1: Breakdown of time periods

mornings	morning	afternoon	in the evening
5:00 - 8:59	9:00 - 11:59	12:00 - 17:59	18:00-4:59 (next morning)

(2) Publishing dynamics

In this paper, a dynamic refers to what a user is doing about the moment. Users can post dynamics to the system to provide other users who are browsing the dynamics with a reference of what they are doing at the moment; at the same time, interested users can also like and comment on the dynamics. Additionally, if the user is a business owner, they can post special offers or holiday benefits for the time period.

(3) Dynamic Recommendations

Dynamic recommendation is a key functional module of the system, the system will be based on the user's time period and the user's historical behavioral data for temporal recommendation. Users can browse the dynamic recommendations of the current time period, and can also interact with the interested dynamics by commenting, liking, forwarding and so on.

(4) Classification of actions

In order to facilitate users to arrange actions or activities according to their own time, the dynamic content in the system is categorized and managed, and the dynamics of the same time period are put together. For example, when the user selects the time period of 5:00-8:59, the system will present the dynamics related to morning jogging, morning reading, eating breakfast, etc.; in this way, users who have the need for time planning can directly click on the specific time period to view.

(5) Search function

In a personalized action recommendation system, the search function allows users to perform fuzzy searches based on keywords to find dynamic content that matches their interests or needs.

(6) Extraction of action recommendations

Extractive action recommendation refers to only one recommendation extracted by the system in a fixed time period, and only one relevant result is recommended to the user. This design can effectively avoid too much recommendation interference and information repetition, and provide a good sense of platform experience for users who have difficulty in choosing.

(7) Supervision by administrators

The whole system operation process, in addition to user participation, but also need to administrator supervision and management, review each dynamic content, the violation of the platform discipline of dynamic content deleted, in order to ensure that the system normal operation of the order.

3.2 Summary of requirements

In view of the above demand analysis, it is determined that the system function module is divided into user module and administrator module. Among them, the main functions based on the user are "registration and login", "publish dynamic", "dynamic recommendation", "search function The main functions based on the administrator are "registration and login", "user management", "dynamic

content management", "dynamic content management", "user management", "dynamic content management", "dynamic content management", "dynamic content management", "dynamic content management", "dynamic content management", "dynamic content management", "dynamic content management", "technical support". Figure 1 is the functional framework of the system.

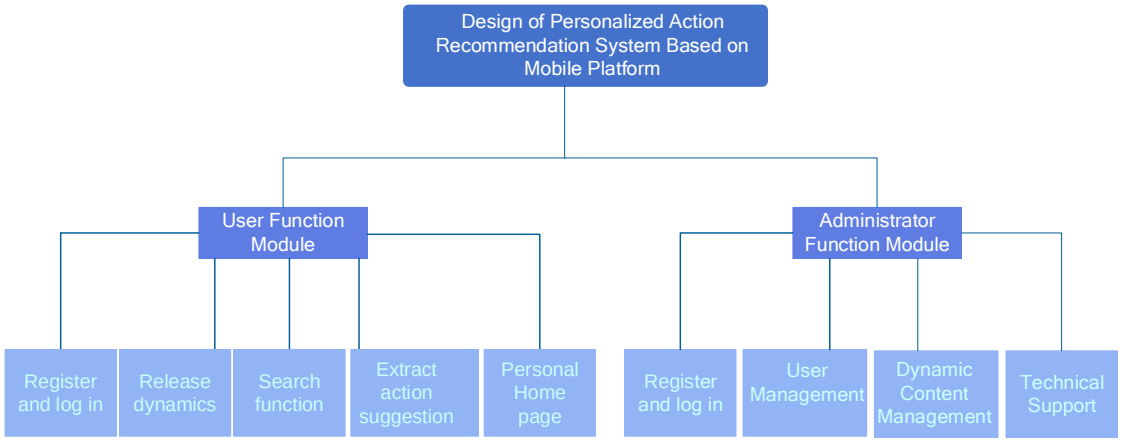

Figure 1: Functional framework diagram of personalized action recommendation system based on mobile platforms

4. Overall Design of Personalized Action Recommendation System

4.1 Personalized Action Recommendation System Architecture Design

Through the requirement analysis of personalized action system, it is known that the system needs to process a large amount of user history data. In this regard, the article adopts the C/S (client-server) architecture as shown below, which puts the computational tasks on the server side to make full use of the server's computational power and improve the recommendation efficiency. The system architecture diagram of this article is divided into three layers, which are representation layer, logic layer and data layer, as shown in Figure 2.

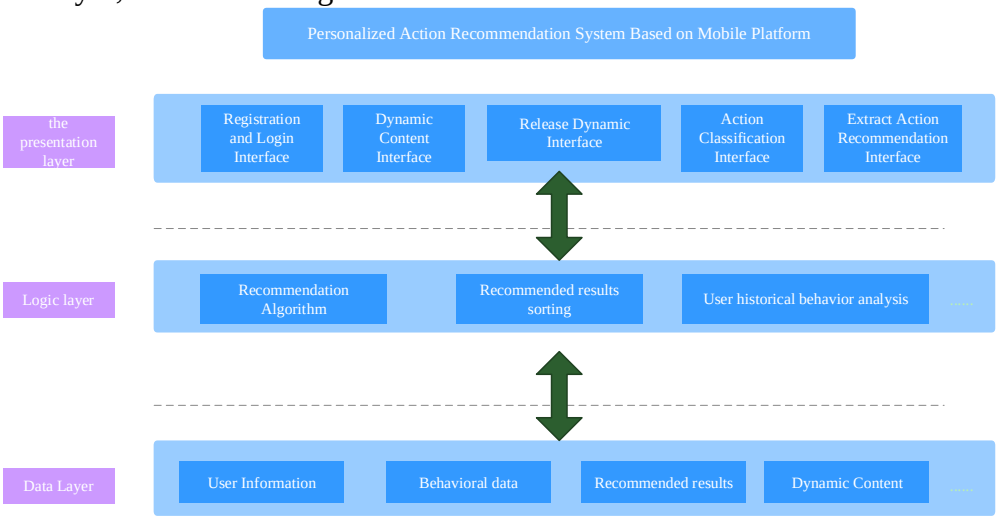

Figure 2: Schematic diagram of the framework of personalized action recommendation system based on mobile platforms

(1) Representation layer

The representation layer is responsible for the front-end page of this system, which provides users with an interactive interface on the mobile platform, including the registration and login function, dynamic recommendation function, publish dynamic function, and extract action recommendation function. The user's interactive operations on the interface will stimulate corresponding commands and forward them to the logic layer for processing.

(2) Logic layer

The logic layer plays a key role in the action recommendation system, which is responsible for analyzing the user's historical behavioral data, such as clicks, views, likes, etc., generating personalized action recommendation results for the user, interacting with the data layer and obtaining the user's historical data. At the same time, this layer uses recommendation algorithms to generate personalized recommendation results for the user, and uses sorting algorithms to generate the final recommendation list.

(3) Data layer

The data layer of this system uses a database (MySQL) to store and manage various data. It is mainly responsible for storing, accessing and processing the data to provide the data needed by the logical and representation layers. In this case, the data is organized in the form of tables, and each table stores a specific type of data, e.g., user information, user behavior data, recommendation results, etc.

4.2 Personalized Action Recommendation System Database Design

In order to be able to store a large amount of data of the system, a personalized action recommendation system cannot be separated from an efficient and reliable database. Figure 3 shows the E-R diagram of the system database, which well demonstrates the interrelationships between the entities.

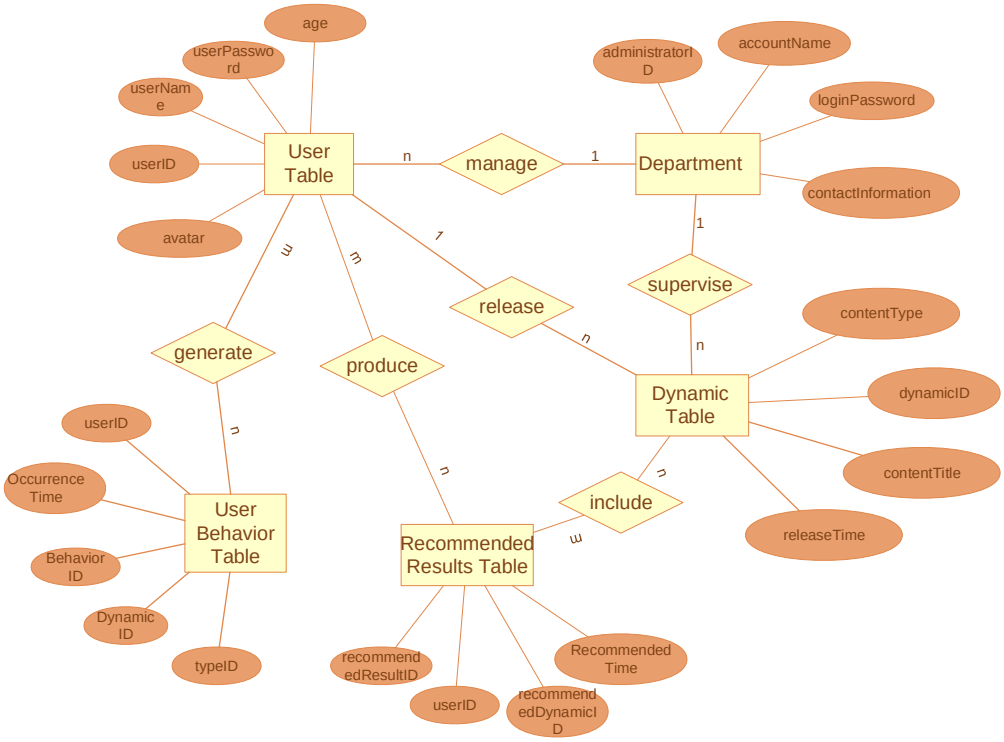

Figure 3: E-R diagram of the database of a personalized action recommendation system based on a mobile platform

Meanwhile, according to the above E-R diagram, the personalized action recommendation system mainly includes the user table (User), the administrator table (Administrator), the user behavior table (UserAction), the recommendation result table (Recommendation), and the dynamic table (Content), etc., as shown in Tables 2 - Table 6.

Table 2: User table (User)

Field name	Field type	Field length	Field Description
UserID	char (computing)	10	User login ID (primary key)
UserName	char (computing)	10	User ID
Password	char (computing)	10	User Login Password
Avatar	blob	-	User avatar
Age	char (computing)	10	User age

The User table (User) is used to store basic information about users. Each user has a uniquely represented ID (UserID), as well as several other fields, which will be used in the login and personalized recommendation process. Among them, the age (Age) is used for later data analysis and statistics. Also, the User table will be associated with the User Behavior table.

Table 3: Administrators Table (Administrator)

Field name	Field type	Field length	Field Description
AdminID	char (computing)	10	Administrator login ID (primary key)
UserName	char (computing)	10	account name
Password	char (computing)	10	login password
Contact	char (computing)	20	contact details

The Administrator table is used to store basic information about system administrators. Each administrator has a uniquely represented ID (AdminID), including user name, password, contact information and other attributes. This information can be used for administrators to log in to the system and have the authority to manage and operate the system. Administrators can add, delete or modify user information, handle faults and other operations to maintain system stability.

Table 4: Table of User Actions (UserAction)

Field name	Field type	Field length	Field Description
ActionID	char (computing)	10	User behavior ID (primary key)
UserID	char (computing)	10	User ID (foreign key)
ContentID	char (computing)	10	Dynamic ID (foreign key)
ActionType	char (computing)	10	Types of user behavior (e.g., comments, likes, etc.)
CreatedTime	datetime	-	Time of User Behavior

The UserAction table is used to record user behavior data. Each record contains a unique behavior ID (UserID) associated with the corresponding user ID (UserID) and dynamic ID (ContentID). This table allows you to analyze the user's behavioral data and understand the user's preferences.

Table 5: Table of Recommendation Results (Recommendation)

Field name	Field type	Field length	Field Description
RecommendationID	char (computing)	10	Recommended Result ID (Primary Key)
UserID	char (computing)	10	User ID (foreign key)
ContentID	char (computing)	10	Recommended Dynamic ID (Foreign Key)
CreatedTime	datetime	-	Dynamic recommended time

Recommendation table (Recommendation) is used to store personalized recommendation results.

Each record contains a unique recommendation result ID (RecommendationID), associated with the corresponding user ID (UserID) and recommended dynamic ID (ContentID). Through the recommended results associated with the user, the system can generate a personalized recommendation list for each user and adjust the recommended content according to the time period.

Table 6: Dynamic Table (Content)

Field name	Field type	Field length	Field Description
ContentID	char (computing)	20	Dynamic ID (primary key)
Title	char (computing)	20	Content Title
ContentType	char (computing)	10	Type of content
CreatedTime	datetime	-	Release time

The dynamic table (Content) is used to store information about dynamic content. Each content is identified by a unique dynamic ID (ContentID), including the title, type and publishing time of the dynamic content. Among them, the dynamic content can be in the form of articles, videos, images, etc.

4.3 Algorithm Design of Personalized Action Recommendation System

Recommendations in a personalized action recommendation system are based on time period and content (historical user behavior data). The specific steps are as follows.

(1) Observation Matrix

When a user makes a new action and publishes a new dynamic, the database updates the data stored in the UserAction table and the Dynamic table (Content) respectively, and forms a new observation matrix A and observation matrix B. Observation matrix A consists of two variables: the type of the user action, and the type of the dynamic content operated by the user. And observation matrix B consists of dynamic content type and dynamic content title. Taking the user behavior with ID 2120012024 and its published dynamic (dynamic ID 20240121095416) as an example, the observation matrices are formed as shown in Table 7 and Table 8.

Table 7: Observation matrix (A)

Types of User Behavior	Types of dynamic content manipulated by the user
commentaries	traveler's tour
commentaries	restaurant visitors
strike (on the keyboard)	traveler's tour
kudos	life-style
...	...

Table 8: Observation matrix (B)

Dynamic content types	Dynamic Content Title
traveler's tour	Traveling through the ancient capital of a thousand years, exploring the sacred place of Terracotta Warriors and Horses

(2) Cosine similarity

For each dynamic to be recommended, a dynamic observation matrix corresponding to it is generated, which is then combined with the user's historical behavior observation matrix to calculate the cosine similarity between them. Using the cosine similarity formula, the formula is calculated as

$$\cos(A_i, B_j) = \frac{\overrightarrow{A_i} \times \overrightarrow{B_j}}{|\overrightarrow{A_i}| \times |\overrightarrow{B_j}|}, \quad (1)$$

Where. A_i refers to the historical behavioral preference profile of the ith user, andB_j refers to the jth dynamic content to be recommended for the ith user. The closer the cosine similarity is to 1, the closer the dynamic content to be recommended is to the user's preference; conversely, the closer the cosine similarity is to -1, the further the dynamic content to be recommended is from the user's preference [6].

(3) Recommended List

Based on the current time period in which the system is located, all dynamic content is divided into time periods, and only action recommendations for the current time period appear in the dynamic recommendation list. Then the cosine similarity calculated above is utilized for sorting, and the cosine similarity is displayed to the user in order from the largest to the smallest.

5. Conclusions

This paper presents the basic design of a personalized action recommendation system that helps users get the best action recommendations for the current time period. Unlike traditional recommender systems, this system incorporates the time period as a key factor, and then calculates the cosine similarity through the user's historical preference features and dynamic content features to generate a recommendation list to display to the user.

The personalized action recommendation system can greatly improve the user's ability to make decisions about the action of the moment, and it can also drive the market economy through the presence of merchants. In the future, internal users will be invited to upload dynamic content to populate the dynamic recommendation page and serve as the initial content display.

References

[1] Song Yafei, Chang Lingxia, Shi Xiujun. Analysis and Prospect of Mobile Internet Security Technology [J]. Network Security Technology and Application, 2022, (05): 85-86.

[2] Liu Yufang, Wang Shaoqing, Zheng Shun et al. A cross-domain user preference transfer framework for cold-start recommendation [J]. Journal of Shandong University of Technology (Natural Science Edition), 2024, 38(01): 26-32+41.

[3] Qiu Yue, Guo Jia, HUAN Jun. Design and Implementation of Smart city Cross-platform Mobile Application based on Xamarin [J]. Modern Surveying and Mapping, 2021, 44(01): 55-59.

[4] Feng Jun. Perspectives on C# core technologies [M]. Mechanical Industry Press: 202103.284.

[5] Zhao Yipin. Design and Implementation of Bank Knowledge Base Management System based on Spring Boot and MyBatis [D]. Shandong University, 2020.

[6] Yuzhe Wang. Research on content-based movie recommendation algorithm [J]. Information System Engineering, 2023, (12): 117-120.