CHAPTER 3

SYSTEM ARCHITECTURE

3.1 OVERALL SYSTEM ARCHITECTURE

The overall architecture of the system proposed in this research work is shown in Figure 3.1. It consists of six different components namely User interface, Data set, Rule Base, Decision Manager, Frequent Itemset Generator component and finally User Profile Manager model which uses the User Preference Database (UPDB) for fixing the support threshold.

Figure 3.1 Overall System Architecture
3.2 USER INTERFACE

The User Interface component of the proposed system is one of the most important components in the application interface because; a user can interact with the application only through this user interface. This user interface provides facilities for effective communication with Decision Manager. Finally, this user interface is responsible for accepting the user input and to forward them to the Data Set, Rule Base and Frequent Itemset Generator components to process the data.

3.3 DATA SETS

In this work, three types of input data sets have been used for the implementation of the Frequent Itemsets generation. Firstly, the UCI data set (Frank et al 2010 (Diabetic and ICU)) is used as the input data sets for testing the temporal frequent itemset mining algorithms. Secondly, the travel and transaction data set is used as the input data set for validating the frequent itemset generation using the Bit Vector Mining algorithms. Finally, the mining log data is used as the input data sets in order to find the exact support count to generate Frequent Itemsets.

3.4 RULE BASE

The rule base is capable of providing rule matching and rule firing techniques, so that accuracy level is increased. The rules are stored and retrieved from the rule base, which consists of a set of IF … THEN rules. This is an important subcomponent for the decision manager to make effective decisions.
3.5  **DECISION MANAGER**

The overall control of the system is with the decision manager. This decision making module uses a decision agent which performs many functions including the selection of a suitable frequent itemset generation algorithm from the five proposed algorithms (Temporal FP-Tree, Hashing Quadratic Probing, Space Preservation Mining, Enhanced Cluster based Bit Vector Association Rule, Temporal Pattern Mining) present in the Frequent Itemset Generator using production rules. This module uses a set of intelligent agents namely the data set selection agent, decision making agent and action agent to make a final decision.

3.6  **FREQUENT ITEMSET GENERATOR**

It consists of two sub components namely Data Structure based and Bit Vector Mining based itemset generator. The Data Structure component consists of two sub modules namely Temporal FP-Tree construction sub module and the Hashing Quadratic Probing (HQP) sub module. The Bit Vector Mining component consists of three sub modules namely Space Preservation Mining (SPM) sub module, Enhanced Cluster based Bit Vector Association Rule sub module and Temporal Pattern Mining sub module respectively. All these five sub modules are capable of generating frequent itemsets from very large databases as well as from temporal databases.

3.6.1  **Data Structure based Frequent Itemset Generation**

Frequent itemsets are generated using the standard data structure component by two ways where, the first one uses the Temporal Frequent Pattern Tree algorithm and the second one uses hashing with Quadratic Probing for effective data mining. These mining methods are used to find
frequent itemsets from temporal databases and conventional transaction databases respectively.

3.6.1.1 Temporal Frequent Pattern Tree Algorithm

In this work, an efficient tree based algorithm called Temporal FP-Tree algorithm with a FP-Tree structure has been developed to mine frequent patterns, conditional pattern bases and sub-conditional pattern tree recursively. This algorithm is used to mine frequent patterns from temporal data. Moreover, the Temporal FP-Tree is constructed dynamically in order to perform effective decision making based on past and present data.

3.6.1.2 Hashing with Quadratic Probing Algorithm

In this work, a new association rule mining algorithm called Hash Based Frequent Itemsets-Quadratic Probing in which, hashing technology has been proposed and implemented to store the database in the vertical data format and to avoid hash collision and primary clustering problem in hashing. Quadratic probing technique utilises the vertical hashing method. The advantage of this new hashing technique is to ease computation of the hash function, fast access of data and processing efficiency. This algorithm provides facilities to avoid unnecessary scans to the database. HBFI-QP does not use all the bins and hence, the phenomenon of primary clustering does not occur with quadratic probing.

3.6.2 Bit Vector Mining Algorithm

In this model, bit vectors are introduced to enhance the performance of rule mining. Based on bit vectors, three algorithms have been proposed in this work. Among them, the first method is the basic bit vector matrix representation; the second is the Enhanced Cluster based Bit Vector
Association Rule Mining and the third is the Bit Vector Temporal Pattern Mining algorithm.

### 3.6.2.1 Bit Vector Matrix Representation

In this model, a bit vector $BV_i$ with $i = 1 \ldots n$ is used for storing the bits where $n$ is the number of transactions in database D. In this representation, if an item $i$ is present in the transaction, it is represented as 1 else as 0. The count of each matrix value is expressed in this work in the form of a three dimensional matrix. This matrix is an upper triangular matrix and thus redundancies of itemsets are avoided.

### 3.6.2.2 Enhanced Cluster based Bit Vector Association Rule Mining

This algorithm scans the database of transactions only once to build the clustering table as a two dimensional array where the columns represent items and the rows represent transactions. The count (CNT) represents the number of similar transactions and it consists of bits (0 or 1). The cluster table now consists of bit vectors for all individual items. The 1’s in each column represent the number of occurrences of an item in the data set. The number of 1’s multiplied by the total number of items gives the support threshold of each item.

### 3.6.2.3 Temporal Pattern Mining

In this work, a new algorithm called Temporal Pattern Mining has been proposed to find the frequent temporal pattern based on Clustering, Bit Vector and Variable Threshold. This algorithm incorporates clustering, bit vector and variable threshold during mining. This algorithm reduces the number of computations in the post-processing phase and also provides more flexibility in the process of effective decision making in many applications.
3.7 USER PROFILE MANAGER

If the minimum support value is very small then very large frequent items are discovered and hence, may find many irrelevant items also. On the other hand, if the minimum support value is set with high values then very few frequent itemsets are found by the mining algorithm. This may lead to the omission of some important knowledge. Therefore, the setting of the minimum support is a difficult task in the mining world. Hence, this research work proposes, a user preference database that helps the user to fix the support value range.

3.7.1 Interesting Measure Identification

This module identifies attributes which are available in the user query. Moreover, these identified attributes are checked with the existing user preference query for matching. In this process, attribute and value matching is carried out to identify the interesting attributes.

3.7.2 Interesting Measure Estimation

Through the similarity identification of queries, the top N similar queries from the user preference database are obtained. The final range of minimum support to be offered to the user is computed using this measure and the result will be the favorable range of minimum support that is closer to the user’s mining intention.

3.7.3 User Preference Database

The user preference database maintains frequently used mining queries extracted from the mining log. The system finds most similar queries to the user’s mining intension, aggregates them and obtains the favorable support range for the user.
3.8 CHAPTER SUMMARY

The major contributions of this thesis with respect to the architecture are:

- Proposal of decision manager for integrity user interface with rest of the system and to make effective decisions.

- Introduction of two groups in frequent itemset generation.

- Development of user profile manager for fixing the support threshold.