**CHAPTER: 4**

**IMPLEMENTATIONS**

This chapter of the report describes the Functions, packages and API used in the project.

**MySQL:**

MySQL is an open-source relational database management system. It was originally produced under the GNU General Public License, in which source code is made freely available. MySQL is very popular for Web-hosting applications because of its plethora of Web-optimized features like HTML data types, and because it's available for free. It is part of the Linux, Apache, MySQL, PHP (LAMP) architecture, a combination of platforms that is frequently used to deliver and support advanced Web applications. MySQL runs the back-end databases of some famous websites, including Wikipedia, Google and Facebook- a testament to its stability and robustness despite its decentralized, free-for-all philosophy.

**SWING:**

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes – an API for providing a graphical user interface for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit.Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

**Java Database Connectivity** (**JDBC**)**:**

It is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

**4.2 Functional Modules**

The functional modules included in the project are listed below.

**Insert Module:**

This module provides the functionality of collecting the required data from the designed interface and transmitting it to the appropriate table present in the database designed for this project. If the provided data does not satisfy the given constraints, it must refrain from storing it into the database.

**Update Module:**

This module again has the functionality of collecting the data from the designed interface, but it updates the already existing tuple that matches the provided primary key of the tuple to be updated, by replacing the existing attribute values with the newly collected data. Again, if the newly provided data does not satisfy the given constraints, it must refrain from updating the corresponding tuple.

**Delete Module:**

The delete counterpart is loaded with the ability to delete a single or multiple records from the table. It searches for the tuple, in the query specified table, based on the provided value for an attribute. Admin can delete in the interface, based on which delete module searches for the record corresponding to that provided attribute value and deletes the record.

**Retrieve Module:**

The retrieve module has a basic functionality of accessing the entire specified table from the database and displays it.