



Study of Development of Java Applications in Eclipse Environment and Development of Java Based Calendar Application with Email Notifications

Muhammad Abid Nazir

This thesis is presented as a part of Degree of
Bachelors of Electrical Engineering with
Emphasis on Telecommunications.

Blekinge Institute of Technology

December 2012

Blekinge Institute of Technology
School of Engineering
Department of Electrical Engineering
Supervisor: Muhammad Shahid
Examiner: Dr. Benny Lövström

Abstract

Eclipse is one of the mostly used software in professional development of programming applications and software solutions. It is open source software and provides extensive availability of free libraries.

In this thesis work, Eclipse was studied for Java applications development. To enhance the study and to get hands on experience over Eclipse IDE, an application was developed using Java programming language. The proposed application is a desktop application that can be used on all modern operating systems. Application was developed using Java SE (standard edition) version 1.7, which is the latest version available from Oracle Corporation.

Java Swing API has been used for building GUI (graphical user interface) of the application. Database for event credentials was developed by MySQL database management system. The connection between application and database has been done through Java database connectivity JDBC. Some additional Java APIs were loaded to Eclipse project workspace, and a comprehensive explanation has been provided on how to use external libraries in Eclipse environment.

Keywords: Java, Eclipse, Java Database Management System,

JavaMail, MySQL, Desktop Applications

Acknowledgment

I would like to say thanks to my thesis examiner Dr. Benny Lövström. I am also much thankful to my thesis advisor Muhammad Shahid for his encourage and support during thesis work.

The discussions with Adnan Ali (Nokia) and Nadeem Khan (Fujitsu) related to Java programming have been very useful.

I am much thankful to my parents, family members and friends for their continuous support and help.

Table of contents

List of abbreviations	08
1-Introducation	10
2-Background	
2.1-Introduction to Java	
2.1.1-what is Java?	12
2.1.2-Current features and specifications	12
2.1.3-Graphical User interface in Java	13
2.1.4-Swing API	14
2.2- Introduction to MySQL	
2.2.1-what is MySQL?	14
2.2.2-Latest Versions of MySQL	14
2.3-Java Database Connectivity (JDBC)	
2.3.1-What is JDBC?	15
2.3.2-Operations of JDBC	15
2.4- Eclipse	
2.4.1-What is Eclipse?	16
2.4.2- Java development with Eclipse	17
3-Design and Implementation	
3.1-Requirements	18
3.2-Design	18
3.3-Create a Project in Eclipse	19
3.4-Implementation	23
3.4.1- Layout of the Application	23
3.4.2-Coding for Graphical User Interface	24
3.5-Creation of Database in MySQL	25
3.6-JavaMail API	25
4-Results and Conclusion	29
5-References	31
6-Appendix	32

List of Abbreviations

API	Application Programming Interface
GUI	Graphical User Interface
DBMS	Database Management System
JDBC	Java Database Connectivity
WWW	World Wide Web
SDK	Software Development Kit
IDE	Integrated Development Environment
J2SE	Java Platform, Standard Edition
App	Application
Email	Electronic Mail
JDK	Java Development Kit
JFC	Java Foundation classes
Jar	Java archive

Chapter 1

Introduction

Calendars are primarily used to identify days, months and years. It informs us about events and activities that are important for social, cultural, religious, academic and professional reasons. For instance, a calendar indicates the days, which are religious, cultural and national holidays. Calendars determine the beginning and closing of business accounting records. In our daily life, we need calendar to identify the deadlines for submission of assignments, projects and to pay utility bills etc. One cannot deny the importance of calendars in daily life, since these are used for appointments, deadlines, meetings and anniversary dates.

Today, in the age of World Wide Web, calendars needed to be customized and updated for their increasing importance and use in our daily life. One needs to be alerted for an event i.e. meeting, appointment or any other important date. Communication through email becomes so common in this age, that, it is necessary to create a calendar application, by which one can be alerted through email.

This thesis work reflects a strong experience of Java programming language, and knowledge of connecting the applications with database. A step by step demonstration of how to do Java project in Eclipse, has been presented. There are certain questions as following that have been answered:

- How to build an application of Java SE, using Eclipse IDE?
- How to trace bugs in an application, when using Eclipse and using built in functions to fix those bugs.
- How to connect a Java application with database in Eclipse.

The thesis work was divided into following different parts.

A calendar GUI had to be created that enable the user to create alert event for a day. To set the alert and details of the event, it will prompt user to enter the following details of the event.

- Name of the event

- Venue of event
- Phone no. of the user
- User Email address, to which the reminder will be sent

The development procedure for the application required the following steps:

- Layout of the user Interface
- Coding for the application
- Creation of database for the application
- Connecting the database with application
- Addition of email functionality in the application

Chapter 2

Background

In this chapter, the theoretical background for Java, MySQL and Eclipse software is presented.

2.1 Introduction to Java

2.1.1 What is Java?

Java is a computing platform and high level programming language, which was developed originally by James Gosling at Sun Microsystems. A team at Sun Microsystems named “green team” started working on Java language project in 1991, which was finally released in 1995. Java has its syntax similar to and derived from C, C++ languages. Unless other languages, in which one has to either compile or interpret the code before use it, in Java we have to compile and interpret code to run it. A Java program can run on the following machines provided that they have Java virtual machine (JVM) installed.

- Windows OS
- Linux
- Solaris
- MacOS

2.1.2-Graphical User interface GUI in Java

A Graphical User Interface (GUI) is a human friendly way to interact with computer applications. A GUI gives an application certain ‘look and feel’. A GUI is built from components, and these are the components with which user interacts, operates and controls the application [2].

A simple example of GUI is given in the Figure1. The Figure shows components of a GUI. Some of these components are Buttons, Combo box, Menus, Title bar. These all are controlled by Java’s Swing API’s JButton, JComboBox, and Jmenu etc. The user interacts with these components to use the application.

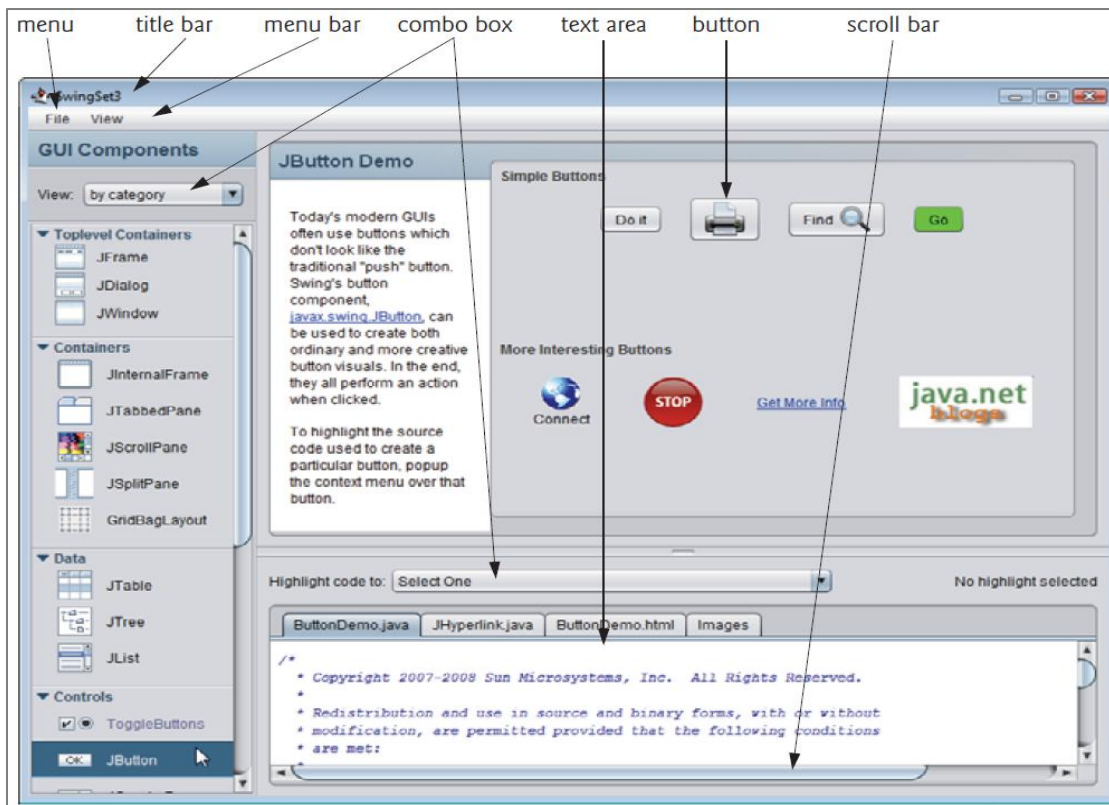


Figure1: Demonstration of graphical User Interface [2]

2.1.3-Swing API

It is Java graphical user interface toolkit. Swing, being a part of Oracle's JFC (Java Foundation Class), is an API which provides graphical user interface for Java programs. The Swing API was developed in success to abstract window toolkit (AWT), and provides enhanced set of GUI components. It also has more sophisticated 'look and feel' and flexible components as compared to abstract window toolkit [3].

To create a Java application with graphical user interface, one has to learn Swing API. Swing allows programmers to create customized look and feel. It includes a larger set of components like buttons, labels, panels and list controls etc.

Following is the list of features which differentiate and make Swing more powerful than AWT.

- One can modify and customize the basic Swing components to the functionality one exactly needs.

- Java 2D API is required if one need to add images, figures, drop shadows, or animations in GUI. It is the Swing that provides this facility; the reason is that Swing is built from 2D package.
- Swing provides data transfer, like copy, paste, cut, drag and drop, within a Java application or within Java and other system applications.
- Swings API also provide an undo and redo facility to developers.
- Swing applications can run both on web browsers (as applets) and as standard desktop application.
- One can make application in swing, that can be used and interact with user worldwide, within their own languages [4].

2.2- Introduction to MySQL

2.2.1-What is MySQL

MySQL is the world's most popular and mostly used database management system. It is an open source RDBMS (relational database management system) and popular choice to use in web as well as desktop applications.

The phrase SQL stands for structured query language. The most important thing in SQL is the tables, since the data is saved in the form of tables that makes it easy to access, retrieve and modify the data.

2.2.2-Latest Versions of MySQL

The latest version of MySQL in use is MySQL 5.5. Following are the features which have been added to MySQL 5.5.

- MySQL 5.5 includes thread handling plugin, to allow multiple clients to connect to server and execute statements.
- It provides new capabilities like proxy users and pluggable authentication.
- MySQL has more stability to sustain on multi-core processing units. Since the trend is to increase the number of cores instead of increasing clock speeds, in today's hardware development. This capability enhances the value of MySQL5.5 to take

advantage of multiple CPUs. Instead of 'wait for CPU to get faster' in conventional processors.

- It provides enhancement to table partitioning.
- MySQL supports IPv6.

2.3-Java Database Connectivity (JDBC)

2.3.1-What is JDBC

It is a Java application programming interface (API), which allows the programmers to access the database management system from within the Java program. JDBC has been developed by JavaSoft, Sun Microsystems.

JDBC allows Java programmers to execute SQL statements, and to access the tabular form data in Java code, using a set of APIs and classes written in Java programming language. By using JDBC, one can INSERT, DELETE and UPDATE data in the SQL database through accessing it from Java program [7].

2.3.2-Operations of JDBC

The operation of JDBC consists of following three basic steps:

- The connection is established with a database.
- To send the SQL statements.
- Results are processed.

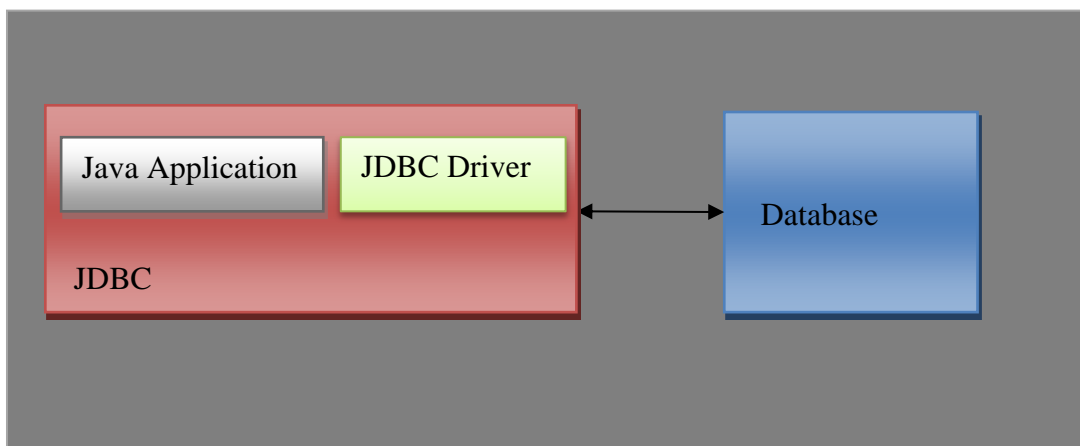


Figure 2: Components of JDBC model [8]

2.4- Eclipse IDE

2.4.1- Introduction to Eclipse IDE.

Eclipse is a multi-lingual software development environment, which consists of integrated development environment (IDE) and extra plug-in system. An IDE provides many different tools e.g. code writing, compiling, running, debugging, file management, and documentation all at one single platform.

Eclipse is one of the IDEs which are mostly used professionally to develop applications and software solutions. An advantage of Eclipse over other professional IDEs is that Eclipse is an open source platform, and therefore, it is easy to add new libraries and resources in it. Most of it is written in Java programming language.

There are a variety of languages, whose applications can be developed in Eclipse by means of extensible plug-in systems. Some of mostly programmed languages in Eclipse are JAVA, C, C++, COBOL, Perl, Android, Python, Ruby, Groovy and Scheme [10].

Below is a Figure that shows typical interactive environment of Eclipse, for the development of an application.

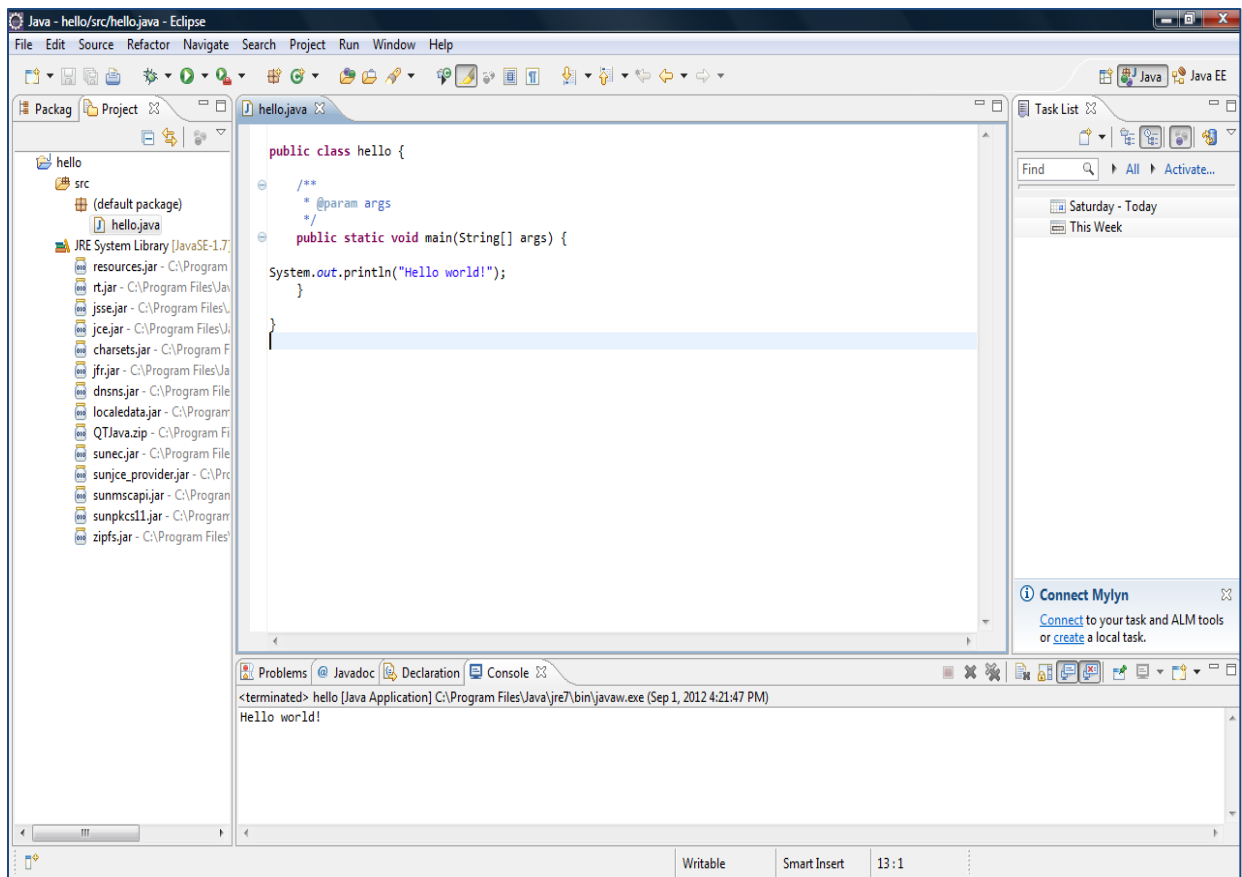


Figure 3: Screenshot of Eclipse development interface

Chapter 3

Design and Implementation

In this chapter different steps for the design and implementation of application are presented.

3.1-Requirements

To develop this application, following softwares and APIs are required:

- Eclipse IDE (Integrated Development Environment)
- MySQL server 5.5
- JDBC connector
- JavaMail API 1.4.5

Eclipse can be downloaded free of cost from internet. A description of how to download, install and configure for use can be found easily. The download link of Eclipse for Java developers has been given in Appendix.

There are mainly two types of editions for MySQL server available. One is MySQL Community edition and other is MySQL Enterprise edition. The difference between them is that the Enterprise edition has more flexibility to choose from multiple MySQL editions for a specific business or technical requirement, and available for 30 days trial. But we will use community edition, because it is freely available version to download and fulfill the requirements. The link to download the MySQL community server latest 5.5 version is available in the appendix.

The latest version of JDBC connector available is J 5.1.21, this is also available free of cost from MySQL website. The links to download the software package and the instructions for installation are available in appendix.

3.2-Design

There are many different approaches to design the application which depends upon the developer/designer. In this case, it requires a good programming knowledge in Java; especially

the things can be easier only if one has command on Java Swing API, and handling the events in graphical user interface. In addition to this, familiarization to Eclipse makes it easier to find and fix the bugs.

3.3-Create a Project in Eclipse

To launch Eclipse software, double click the `eclipse.exe` file in windows, and `eclipse` in Mac/Unix, in the directory where one have installed the Eclipse.

To start a project, Click File > New > Java Project

The following Figures show how to proceed with Eclipse, when creating a certain Java project.

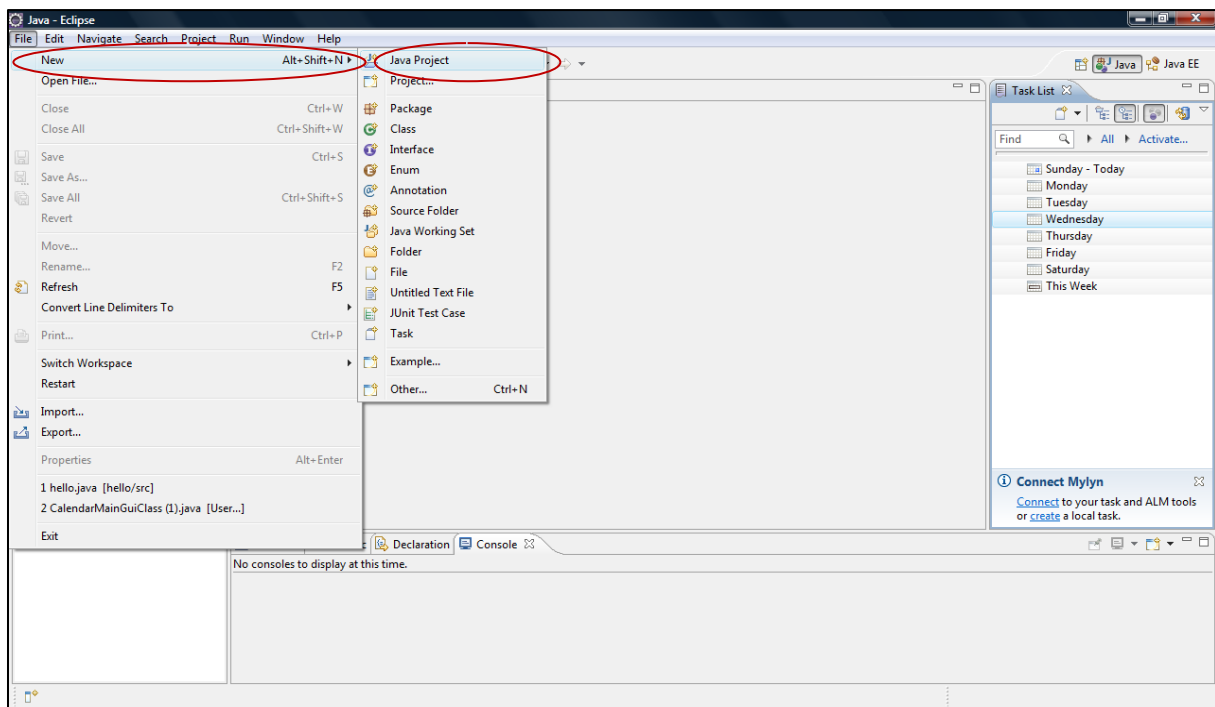


Figure 4(i): Creating a Java Project

Enter the name of the project, and check the option, “*Create separate folders for sources and class files*”. Click finish. The project has been created in the file path (workspace), that one can choose; otherwise it will be saved in the default workspace.

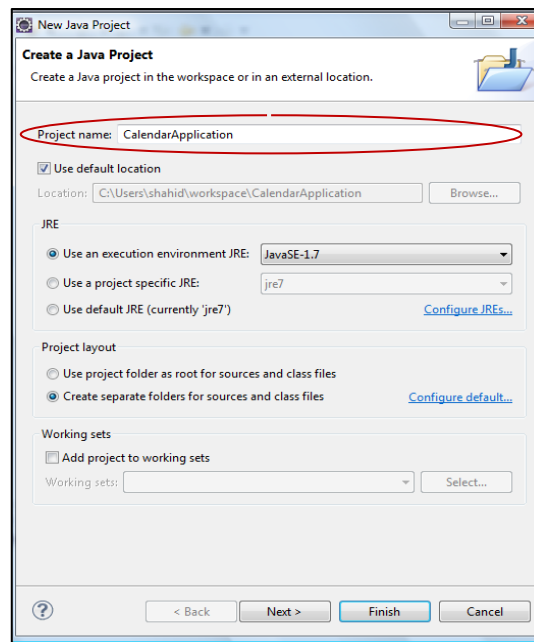


Figure 4(ii): Creating a Java Project

To create a package for this application, click on source folder
Src> New > Package.

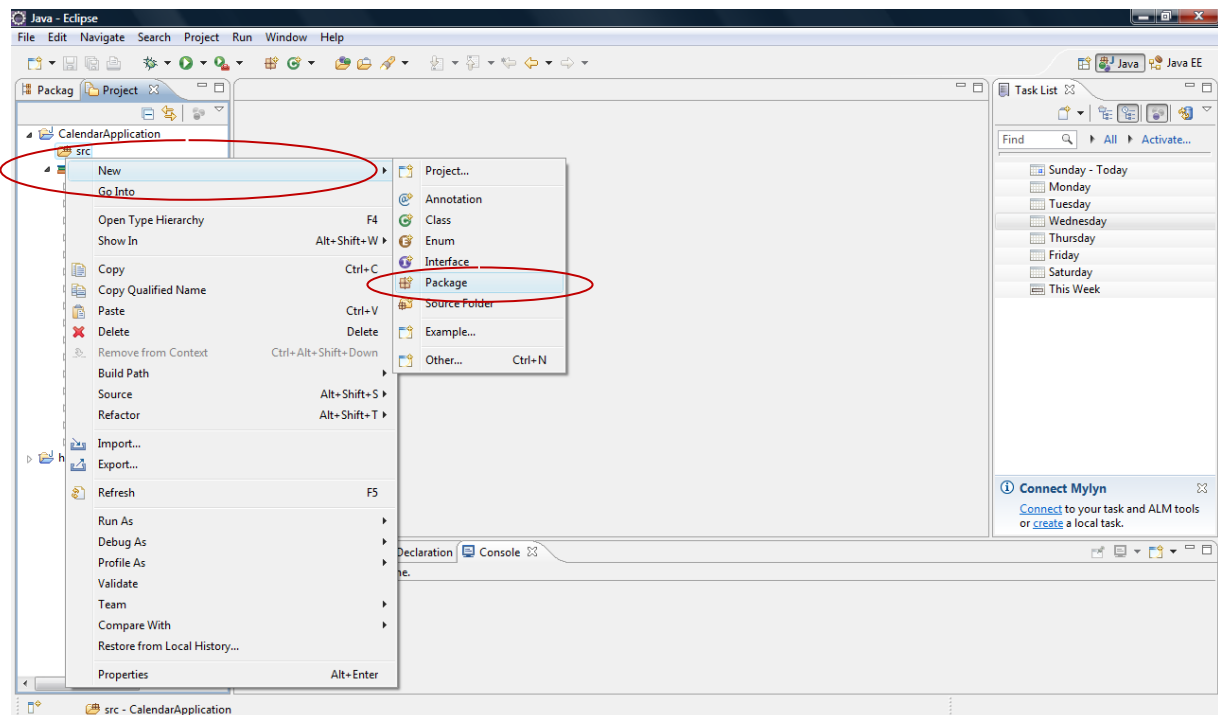


Figure 5(i): Creating a Package in the Java project

Enter the package name and press finish to create the package.

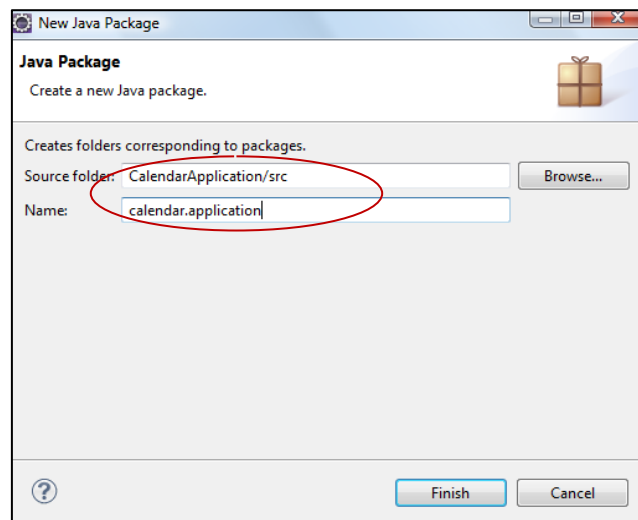


Figure 5(ii): Creating a Package in the Java project

Now to create a new class,

Src > calendar.application > New > Class

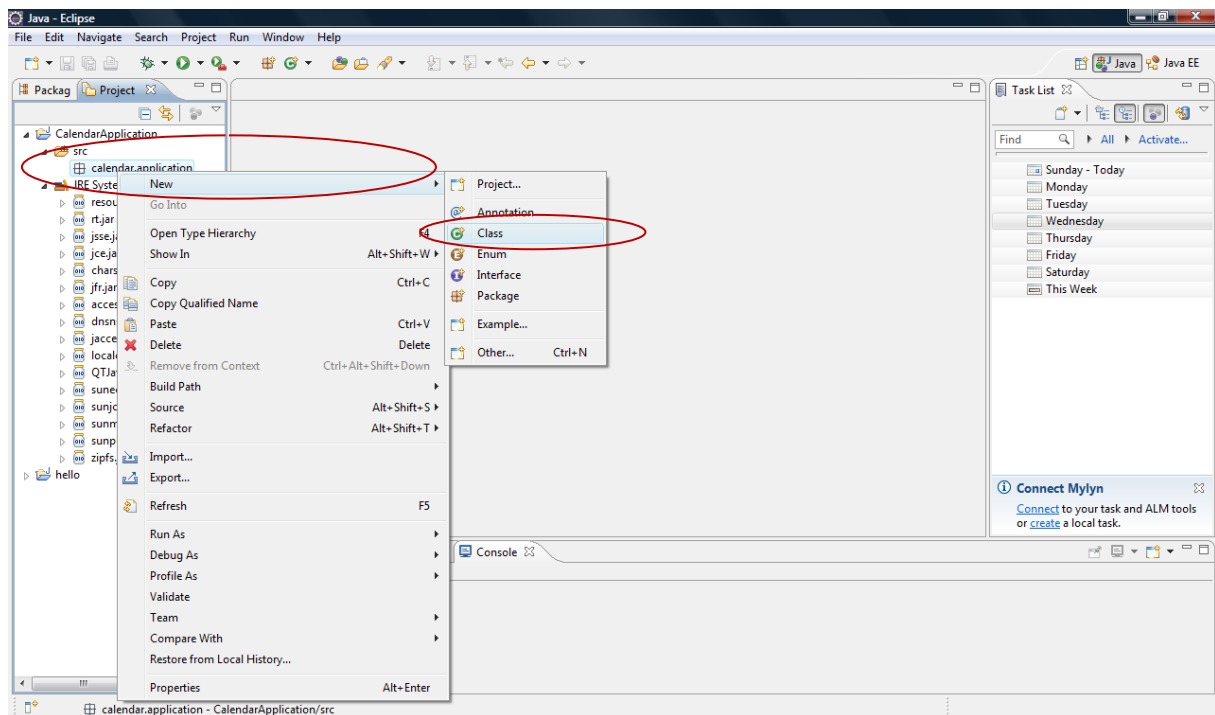


Figure 6(i): Creating a Class in the Java project

Enter the name of the class, it should be noted that, first alphabet of class name must be capital letter.

Click finish to create the class.

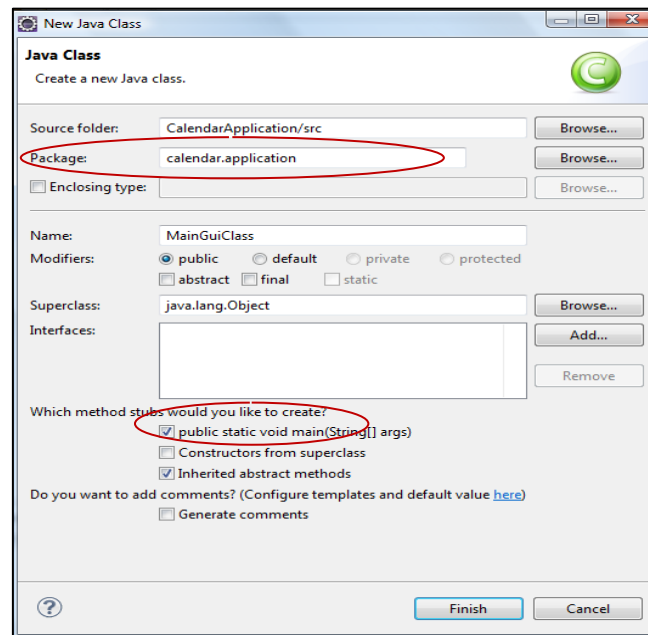


Figure 6(ii): Creating a Class in the Java project

The project has been created and is ready to use. Click on the map name that is given as project name.

CalendarApplication > src > MainGuiClass.Java

We will get to the class, where we can start coding for our application. The class in the initial will look like this:

```
package calendar.application;

public class MainGuiClass {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

3.4-Implementation

Before the implementation of coding in Java, graphical view of the application was first sketched that how it will look like and which components will be needed to add.

There are two graphical windows that are needed for the application. First is the main graphical user interface that appears when the user runs the application. The second graphical window appears when user clicks the add Alarm button to add an event.

3.4.1- Layout of the Application

The following is the layout of window that appears when the user runs the application. It contains a calendar view, days, months and current year. It also includes the Add Alarm button to add and set the new alarm.

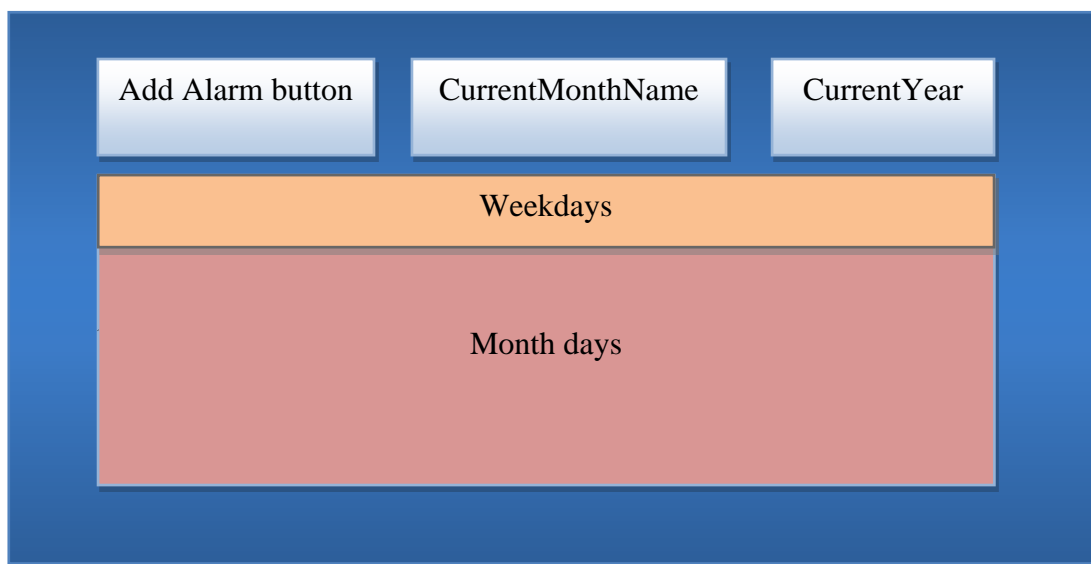


Figure 7: Main Graphical User Interface

The following is the layout of window that appears when the user runs the application to add the alarm. It contains four text fields to add the event name, event venue, contacts phone number and email address to which the email will be sent. It also includes buttons to send, delete, search or update the event.



The image shows a graphical user interface (GUI) for event modification. It consists of a dark blue background with several white buttons and text fields. At the top, there are four stacked text input fields with labels: "Event Name", "Event venue", "Contacts' phone", and "Email adress for Alarm". Below these fields, there are five buttons. On the left, there is a button labeled "Send Mail". In the center, there are four buttons stacked vertically: "Delete event", "Search the event", "Clear event", and "Update". The "Update" button is positioned to the right of the "Delete event" button. The "Search the event" button is positioned below the "Delete event" button. The "Clear event" button is positioned below the "Search the event" button. The "Send Mail" button is positioned to the left of the "Delete event" button.

Figure 8: Interface for event modification

3.4.2-Coding for Graphical User Interface

After creating the project in Eclipse, the next step is the coding for application. Five different classes were created for the application. There was a main class that includes the graphical user interface and event handling of the required functions. The other class was written to

build the connection between main application and database. The third class was written to get the data in the form of strings. The fourth class in Java was written to include methods to add, delete and search the data. There is also a class that used Java mail API, to send email from within the application.

3.5- Database Development in MySQL

3.5.1-Creation of Database in MySQL

To create database for the application, MySQL database management system was used. In MySQL database management system one can create, access and modify the details of the event from within the application. This database contains one table that includes four columns in it, for event name, event venue, contacts phone and receiver email.

3.5.2-Connecting Database with Application

Finally the database and application has been connected by JDBC connector. To do this, one needs *J-connector* that connects MySQL database and Java applications. The jar file of the connector has been imported in the CLASSPATH of the project.

3.6-JavaMail API

JavaMail is an application programming interface that is used to read/compose, send and receive electronic mail messages.

This API is developed by Sun Microsystems. JavaMail API provides protocol independent and platform independent framework for electronic mail transport.

The core classes of JavaMail API are contained in `Javax.mail` and `Javax.mail.activation` packages. JavaMail API can be used for different purposes, e.g., to send a notification at the time of registering, “thanks for interest in my article” message, or sending notification of important updates etc.

JavaMail API can be used for the following major protocols.

- **SMTP-Simple Mail Transfer Protocol:**
It provides mechanism to send/deliver electronic mails.
- **IMAP-Internet Mail Access Protocol:**
Internet mail access protocol is an advanced protocol to receive electronic messages, with multiple mail boxes for a single user.
- **POP3-Post Office Protocol 3:**
It provides mechanism to receive messages. But it supports only single mail box for each user.
- **MIME-Multipurpose Internet Mail Extensions:**
It is used by mail program and not actually the transport protocol. It tells the browser what is being sent, e.g. format of the message and attachment.

Packages and classes of JavaMail API:

There are two core packages for JavaMail API, `Javax.mail` and `Javax.mail.activation`. These packages contain many classes used for Java mail. Some of the main classes which have been used in the proposed application are as follows:

- `Javax.mail.Message`
- `Javax.mail.Session`
- `Javax.mail.Transport`
- `Javax.mail.internet.InternetAddress`
- `Javax.mail.internet.MimeMessage`

Sending Email Using JavaMail API:

There are many different ways to send an email, for that one must have an SMTP server that is responsible to send electronic mails. To avail an SMTP server, one can use the following techniques.

- Installing and using any SMTP server e.g. Apache James server, Postcast server, email server etc.

- Using SMTP server provided by other companies, such as Gmail, Yahoo etc.
- Using SMTP server provided by any host.

In this application, SMTP server provided by Gmail has been used, since the aim was, to send emails from within the application.

Steps in sending an Email:

There are three steps to send an email

- i. Getting session object
- ii. Composing of message
- iii. Sending the message

1- To get the session object

There are two methods to get the object of session in `Javax.mail.Session` class. One is `Session.getDefaultInstance()` and the other is `Session.getInstance()` method.

In this application `Session.getInstance()` method was used. The syntax of the two methods to get the session object using `getInstance()` method is as follows:

```
Public static Session getInstance (Properties      props)
Public static Session getInstance (Properties      props, Authenticator auth)
```

2- To Compose the message

To compose a message one needs to use the class `Javax.mail.Message`. Since this is an abstract class, so, '`Javax.mail.internet.MimeMessage`' class was used because this is most commonly used for composing a message.

One needs to pass session object in `MimeMessage` class constructor to create a message. The syntax of this is as follows.

```
MimeMessage message = new MimeMessage (session);
```

By this, message object was created. To store information in this object, `MimeMessage` class provides many methods. These methods are as follows.

- `public void setFrom(Address address)`
- `public void addRecipients(Message.RecipientType type, String addresses)`
- `public void setSubject(String subject)`
- `public void setText(String textmessage)`

3- To Send the message

`Javax.mail.Transport` class provides the method to send message. These methods are as follows:

- `Public static void send(Message message)`
- `Public static void send(Message message, Address[] address)`

An example to send a message is:

```
Transport.send(message)
```

To send email using JavaMail API, one needs to import two jar files in the project. These Jar files are:

- Mail.jar
- Activation.jar

These files are provided by Sun Microsystems and one can download them free of charge from Oracle's website. The current version of JavaMail API is 1.4.5. The download link has been given in appendix.

Chapter 4

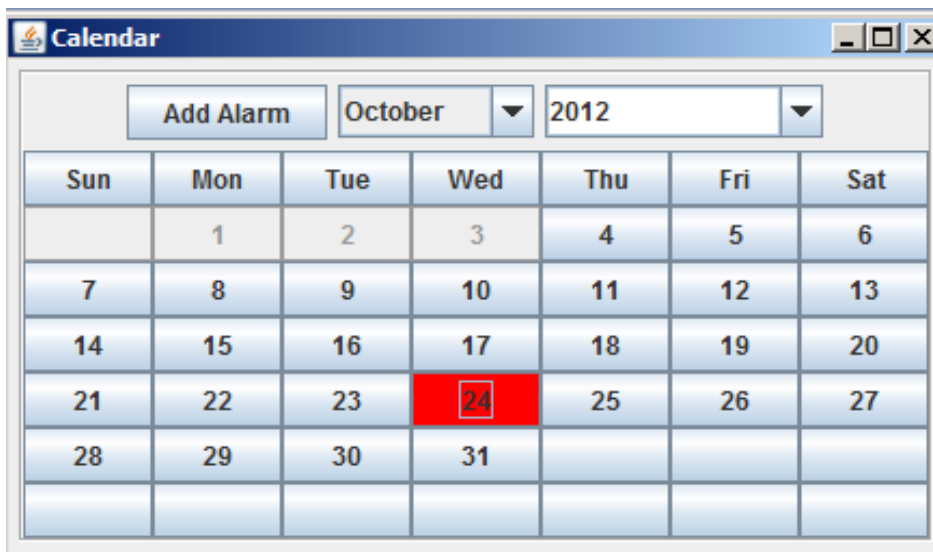
Results and Conclusions

In this chapter the results and conclusions are presented.

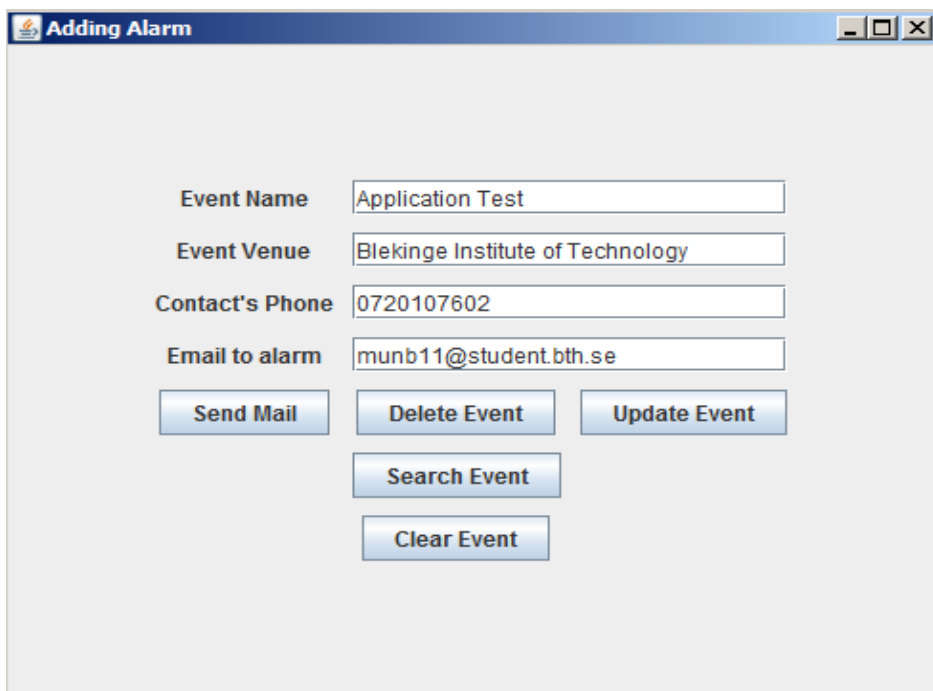
Results:

The graphical user interface of the application is as follows:

Main window:



Graphical window to add notification:



All the proposed functions worked well and the notifications has been sent and checked at different periods of time.

Conclusions:

Application development in Eclipse environment has been explained and the experiment to develop Java desktop application has been done. The tasks which were covered in the study are as follows:

- Step wise explanation of how to develop Java applications in Eclipse IDE.
- Development of desktop calendar application with email alerts in Eclipse IDE.

Eclipse IDE, being a professional-grade development platform has one bigger advantage over other IDEs, that is, it is open source, and it is therefore easy to add new free libraries in it. Experiment to load external libraries in Eclipse project was done successfully. The proposed Java application needed to add external Java archive (jar) files, to make the development easy; in the case the jar files contained packages, which have necessary classes used in the project.

Chapter 5

Appendix

Links to download required softwares

1. Download link for Eclipse:
<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/junor>
[Accessed: 28 Oct, 2012]
2. Download link for MySQL server:
<http://www.mysql.com/downloads/mysql/>
[Accessed: 28 Oct, 2012]
3. Download link for JDBC connector:
<http://www.mysql.com/downloads/connector/j/>
[Accessed: 28Oct 2012]
4. Download link for JavaMail API.
<http://www.oracle.com/technetwork/Java/index-138643.html>
[Accessed: 28Oct 2012]

Chapter 6

References

1- Wikipedia, Java Latest specifications, Wikipedia the free encyclopedia. Aug, 2012 (online)

Available: http://en.wikipedia.org/wiki/Java_version_history#Java_SE_7_.28July_28.2C_2011.29 . [Accessed: 16Aug2012]

2-GUI Components, Paul Deitel & Harvey Deitel, Java How to program 9th Edition. ISBN:0132575663,

Available: <http://www.deitel.com/Books/Java/JavaHowtoProgram9eEarlyObjectsVersion/tabid/3622/Default.aspx>.

3- Wikipedia, Swing, Wikipedia the free encyclopedia, (online) Available: http://en.wikipedia.org/wiki/Swing_%28Java%29. [Accessed: 28July, 2012]

4-Oracle, The Java tutorials. Graphical User Interface, Swing. (Online)

Available: <http://docs.oracle.com/Javase/tutorial/ui/overview/intro.html>
[Accessed:28 July, 2012]

5-Wikipedia, MySQL, (online)

Available: <http://en.wikipedia.org/wiki/MySQL> [accessed: 22 Aug, 2012]

6- Developers MySQL, Documentation. (Online)

Available: <http://dev.mysql.com/doc/refman/5.5/en/mysql-nutshell.html>
[Accessed: 22 Aug, 2012]

7- JDBC, what is JDBC. Roseindia.net. (Online)

Available: <http://www.roseindia.net/jdbc/what-is-jdbc.shtml> [Accessed: 03 Sep 2012]

8- JDBC connection. Work of Java database connectivity. (Online)

Available: <http://aspen.ucs.indiana.edu/webtech/jdbc/overviewpaper/JDBCconn.html> .[Accessed: 03 Sep 2012]

9- Mechanism of JDBC. (Online)

Available: <http://www.datadirect.com/resources/jdbc/faqs/how-does-jdbc-work.html> [Accessed: 04 Sep 2012]

10- Wikipedia, Eclipse Software, Wikipedia the free encyclopedia (Online)

Available: http://en.wikipedia.org/wiki/Eclipse_%28software%29

[Accessed: 11 Sep 2012]

11- Java development tools in eclipse, JDT. (Online)

Available: <http://www.eclipse.org/jdt/> [Accessed: 9 Sep 2012]

12- Developing Java projects in Eclipse. (Online)

Available: <http://www.vogella.com/articles/Eclipse/article.html>. [Accessed: 9 Sep 2012]