HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Lập trình mạng

Giảng viên: TS. Nguyễn Trọng Khánh

Điện thoại/E-mail: khanhnt@ptit.edu.vn

Bộ môn: CNPM- Khoa CNTT1

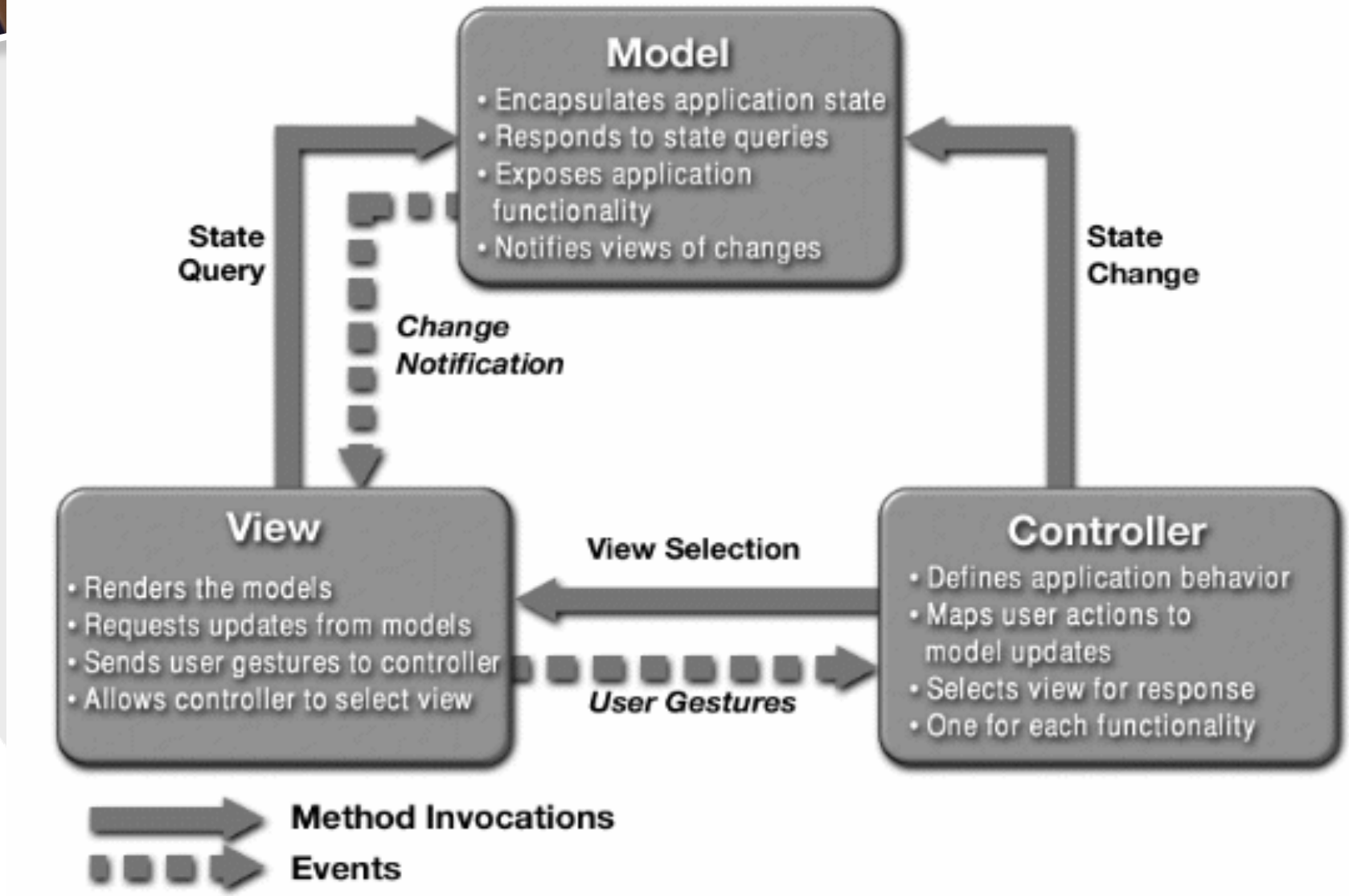
Năm học: August 2021

Thiết kế theo mô hình MVC

Nội dung

- Mô hình MVC tổng quan
- Mô hình MVC cải tiến
- Ví dụ
- Bài tập

Mô hình MVC (1)

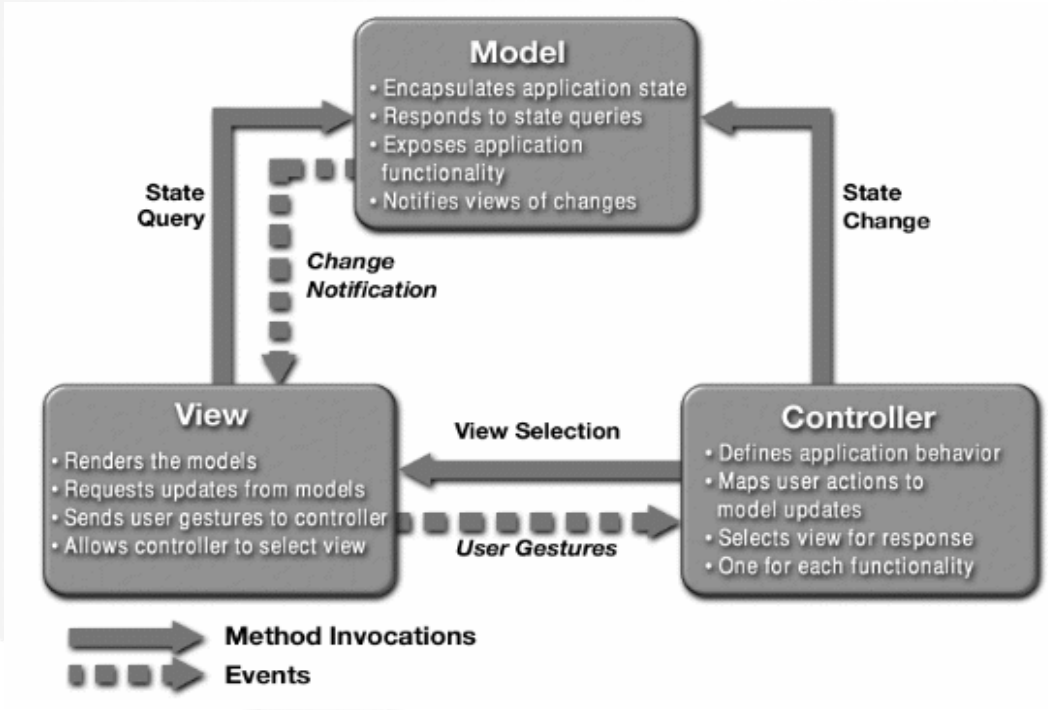


[image source: http://www.oracle.com/technetwork/]

Mô hình MVC (2)

M - model:

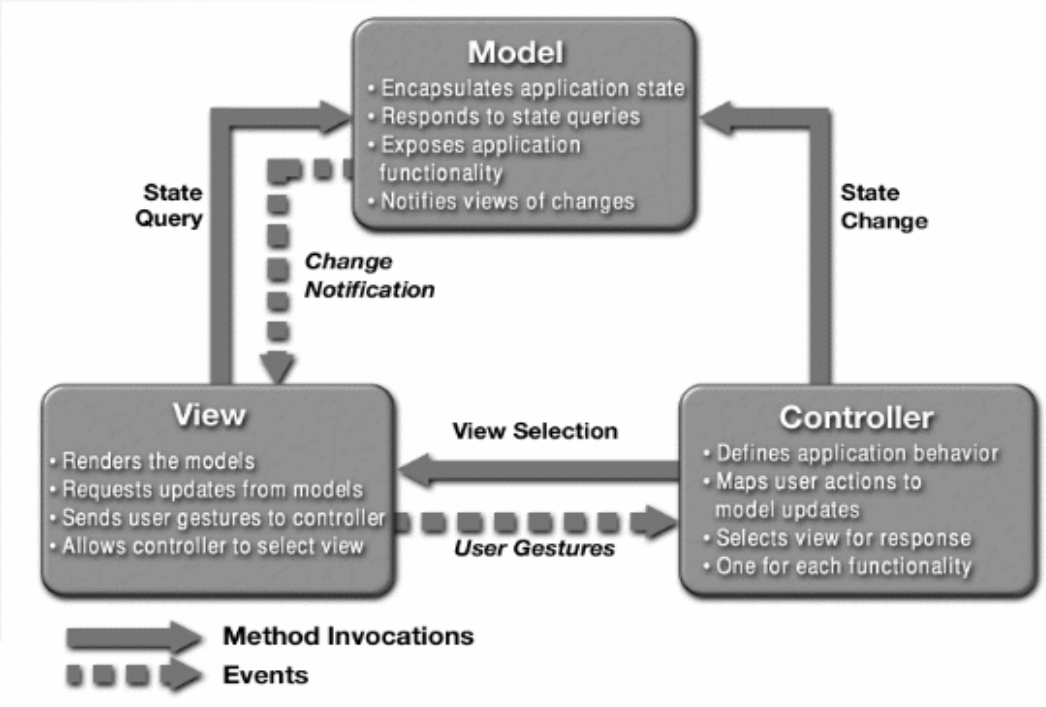
- Đóng gói dữ liệu, thông tin
- Chức năng biểu diễn, vận chuyển thông tin để trình diễn (view) và xử lí (control)



Mô hình MVC (3)

C - control:

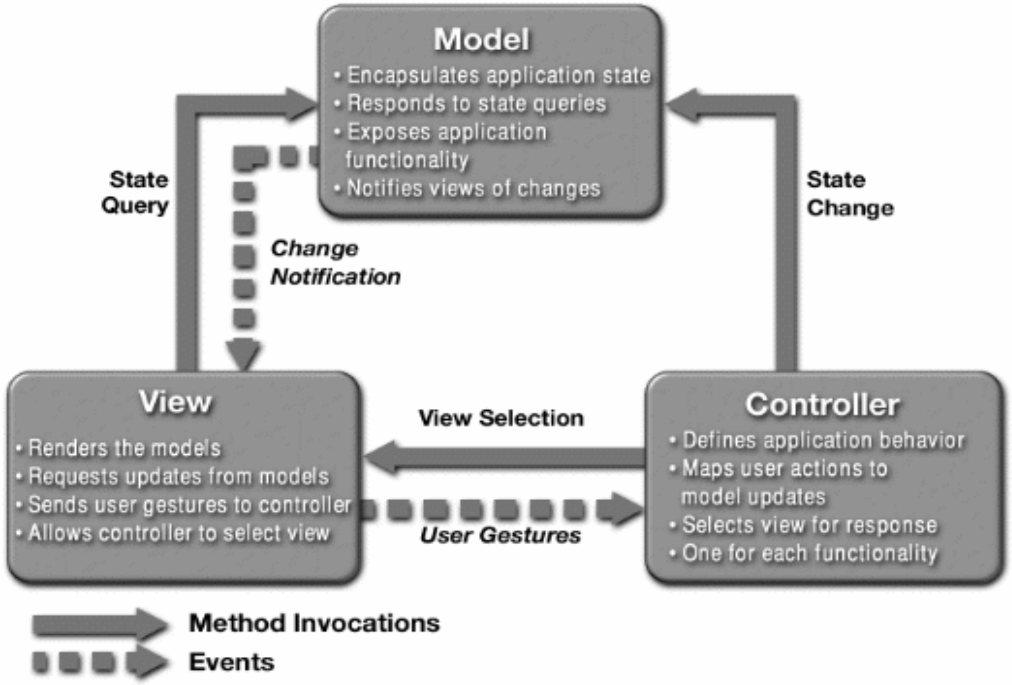
- Định nghĩa các hành vi, hoạt động, xử lí của hệ thống
- Đối chiếu hành động của user (nhận từ view), vào tập chức năng để xử lí, đồng thời chọn hành động đưa view ra để show



Mô hình MVC (4)

V - view:

- Giao diện với người sử dụng
- Show các kết quả xử lí của tầng control
- Thu nhận các hoạt động, yêu cầu của người sử dụng và chuyển cho tầng control xử lí



Thực thi

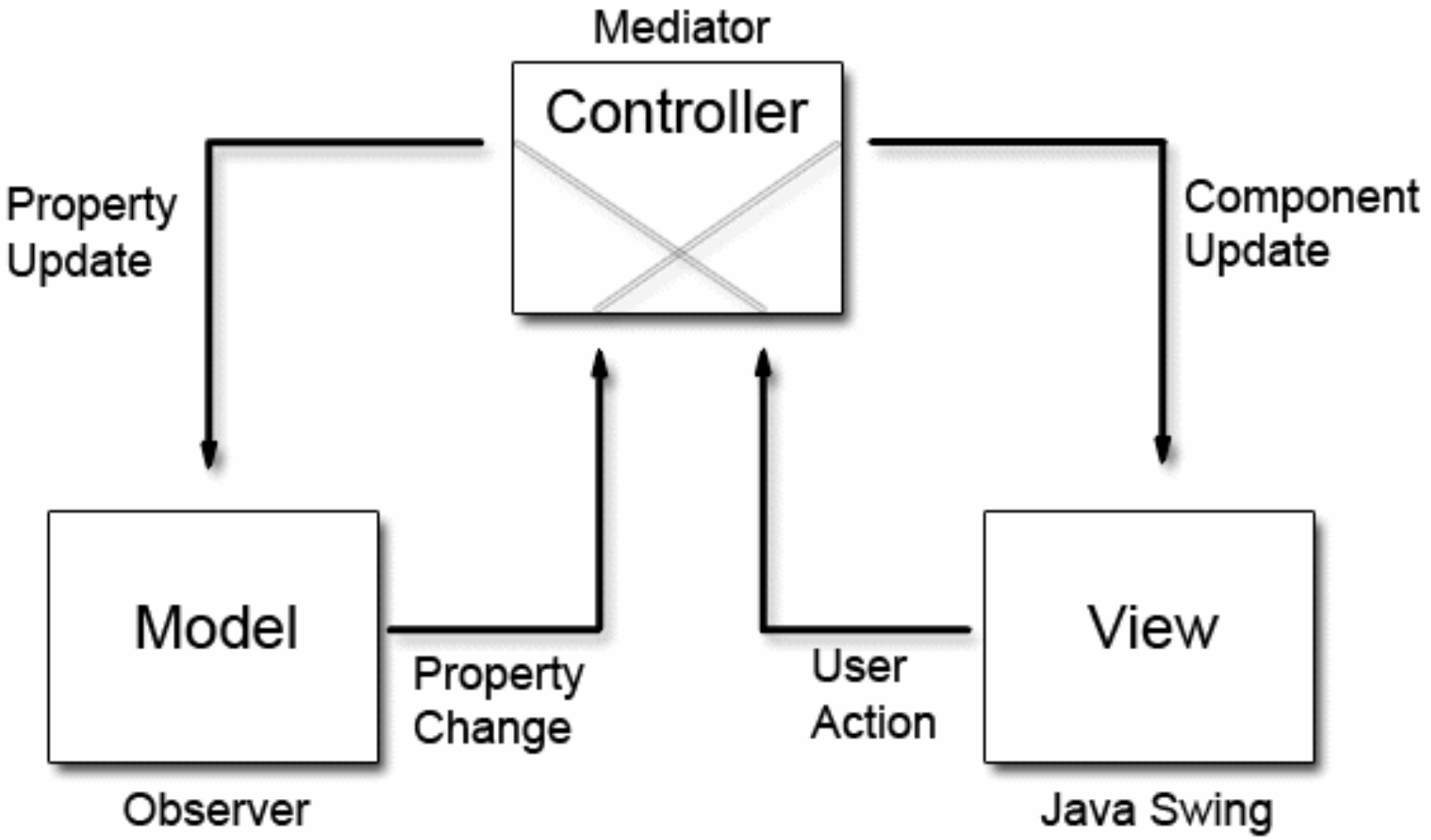
Các bước

- Tạo Model
- Tạo View, và khai báo tham chiếu Model
- Tạo Controller, khai báo tham chiếu Model và View

Hoạt động

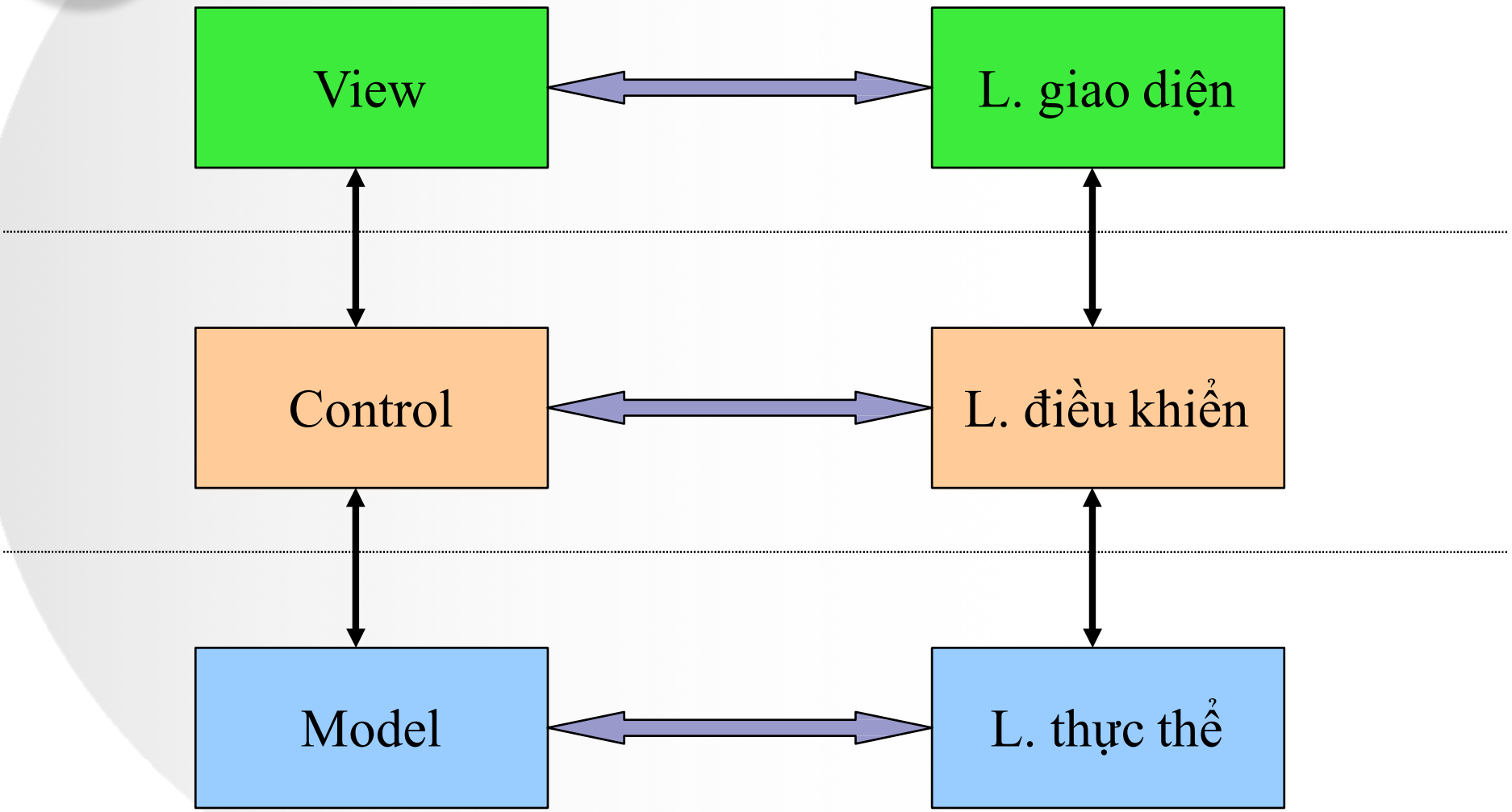
- View nhận sự kiện, ex. click chuột
- View → Controller
- Controller → Model
- Model → View

MVC cải tiến (1)



[image source: http://www.oracle.com/technetwork/]

MVC cải tiến (2)



Các lớp thực thể

- Đóng gói dữ liệu, thông tin
- Chỉ chứa các thuộc tính và các phương thức truy cập các thuộc tính (javaBean)
- Chức năng biểu diễn, vận chuyển thông tin để trình diễn (view) và xử lí (control)

Các lớp điều khiển

- Cập nhật thông tin vào DB (thông tin chứa trong các thực thể)
- Thực hiện các tính toán, xử lí trung gian
- Đối chiếu hành động của user (nhận từ view), vào tập chức năng để xử lí, đồng thời chọn hành động đưa view ra để show

Các lớp giao diện

- Các frame, cửa sổ của ứng dụng (javaSwing)
- Các trang giao diện web: html, jsp
- Các bảng, mẫu biểu, báo cáo in ra

Thực thi

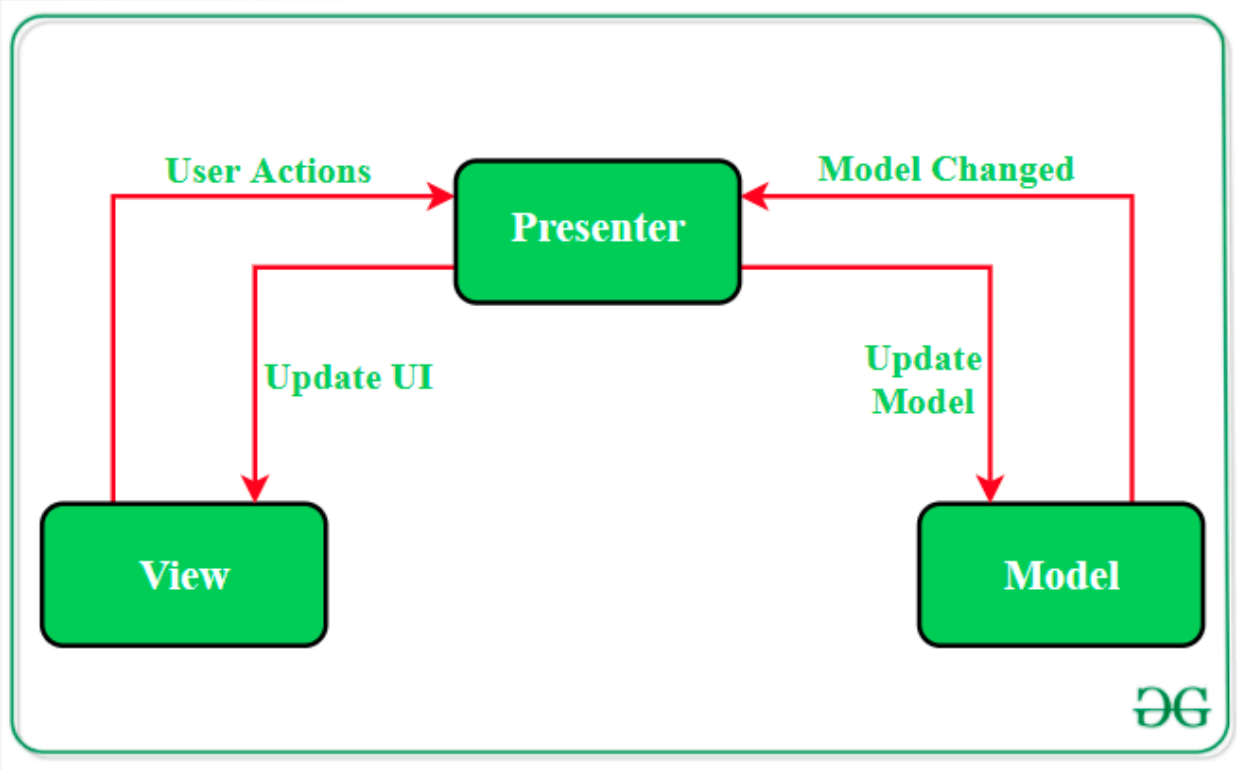
Các bước

- Tạo Model
- Tạo View
- Tạo Controller, khai báo tam chiếu Model và View

Hoạt động

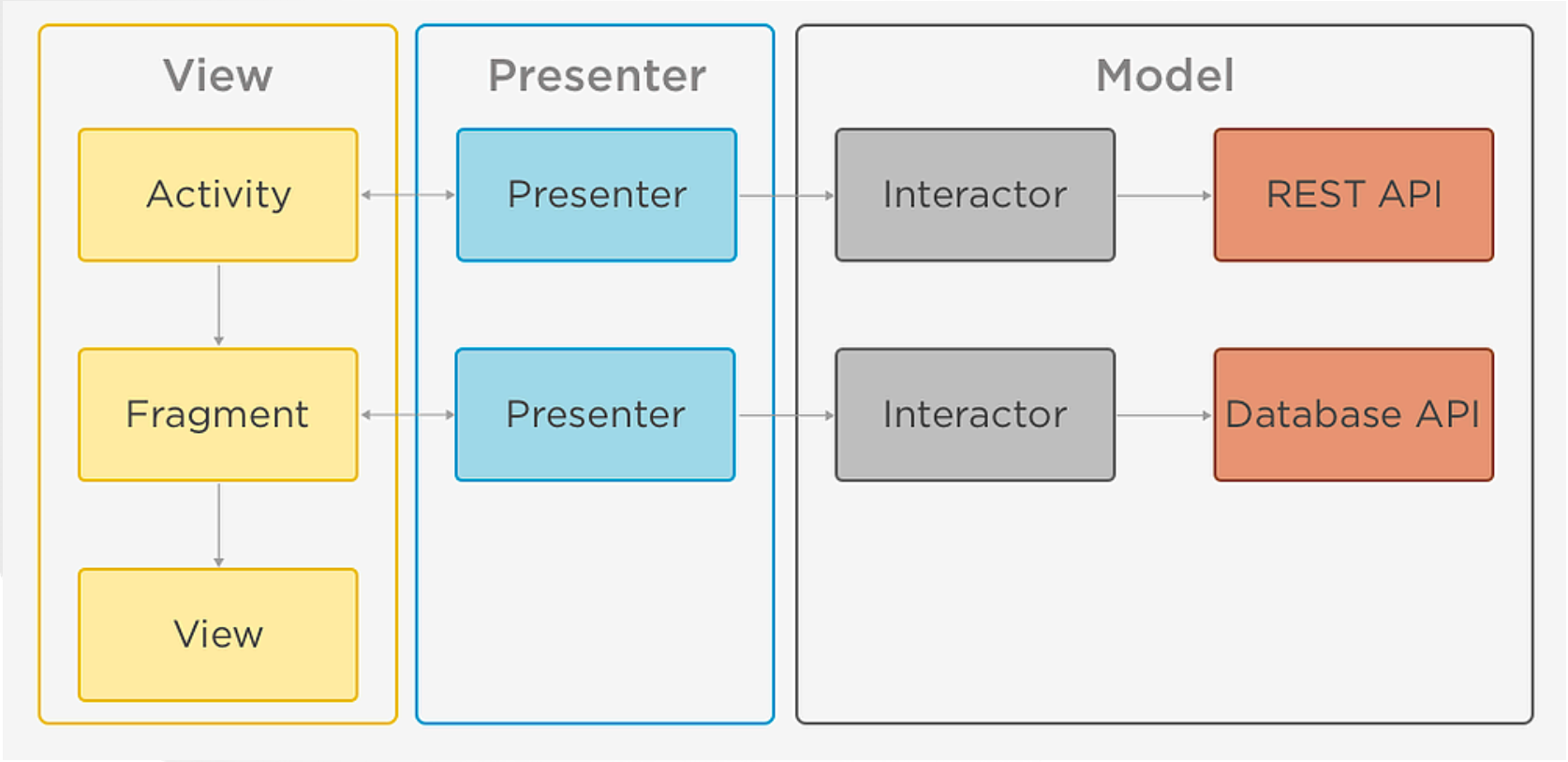
- View nhận sự kiện, ex. click chuột
- View → Controller
- Controller → Model
- Controller → View

Model – View - Presenter



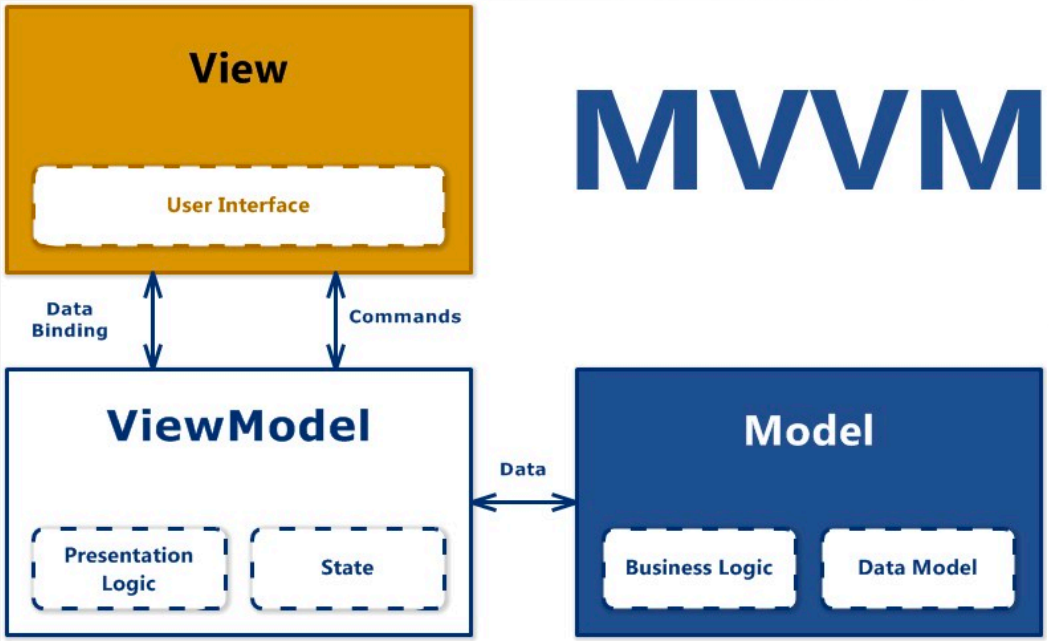
Ví dụ

- MVP trong thiết kế ứng dụng Android

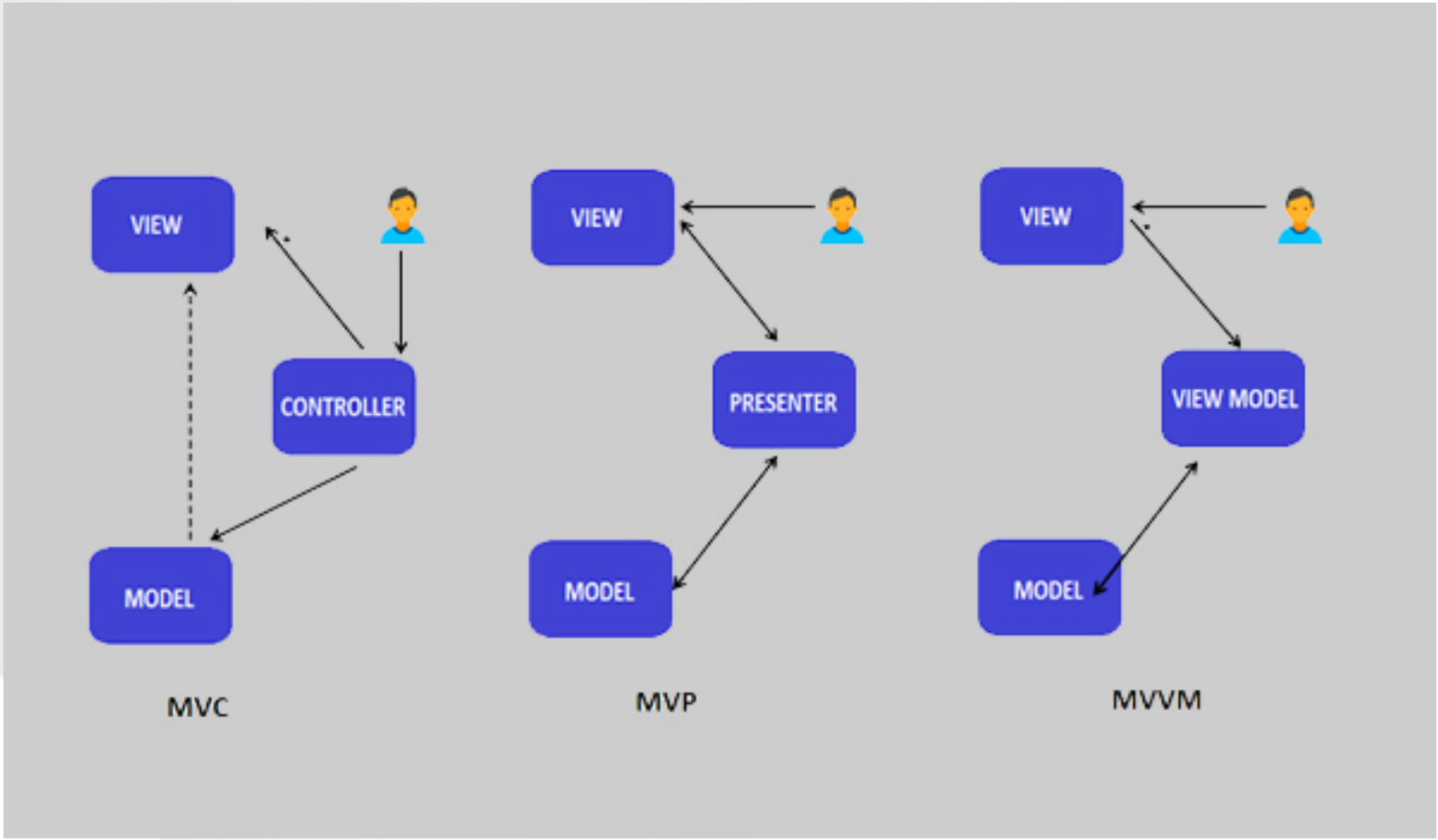


Model View View-Model (MVVM)

- Model
- View
- View-Model: tương tự Controller trong MVC

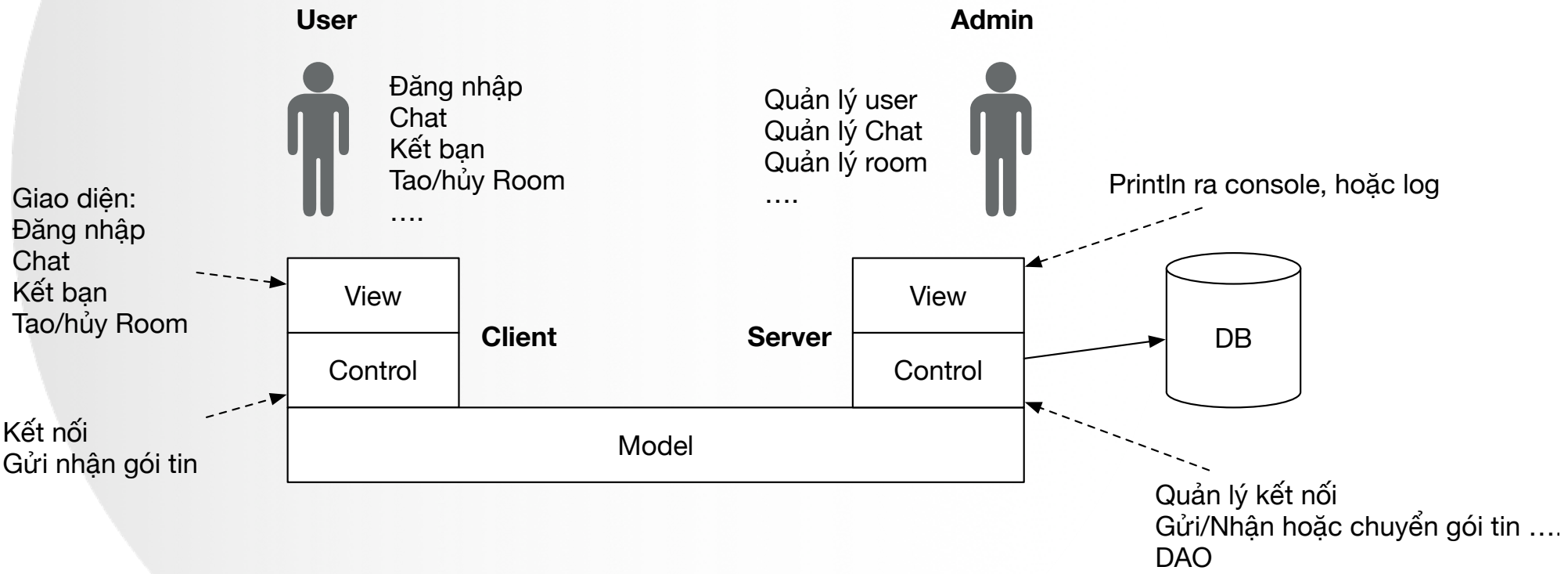


MVC – MVP - MVVM


VIEW
CONTROLLER
MODEL
MVC
VIEW
PRESENTER
MODEL
MVP
VIEW
VIEW MODEL
MODEL
MVVM

Ví dụ

- Ứng dụng chat



Ví dụ:
điều khiển đăng nhập từ dòng lệnh

Login: Model

```
public class LoginModel {
    String userName;
         String password;

public LoginModel(){}

public String getPassword() {
        return password;
}

public void setPassword(String password) {
    this.password = password;
}

public String getUserName() {
    return userName;
}

public void setUserName(String userName) {
            this.userName =
userName;
}
}
```

Login: View (1)

```
import java.io.DataInputStream;
import java.io.IOException;


public class LoginView {
    LoginModel user;

public LoginView(LoginModel user){
    this.user = user;
}

public void showMessage(String smg){
    System.out.println(smg);
}
```

Login: View (2)

```
public void getUserInfo(){
    try{
        Scanner input = new Scanner(System.in);

        System.out.print("Username: ");
        user.setUserName(input.nextLine());
        System.out.print("Password: ");
        user.setPassword(input.nextLine());

        input.close();
    }catch(IOException e){
        System.out.println(e);
    }
}
}
```

Login: Control (1)

```
public class LoginControl {
    LoginModel user;
        LoginView view;

public LoginControl(LoginModel user, LoginView view){
    this.user = user;
        this.view = view;

    while(true){
        view.getUserInfo();
        if(checkLogin()){
            view.showMessage("success!");
            break;
        }else{
            view.showMessage("wrong username or password!");
        }
    }
}
```

Login: Control (2)

```
private boolean checkLogin(){
    if ((user.getUserName().equals("sa"))
        &&(user.getPassword().equals("sa") )){
        return true;
    }
    return false;
}
}
```

Login: main

```
public class LoginMVC {


    public static void main(String[] args){
        LoginModel user = new LoginModel();
        LoginView view = new LoginView(user);
        LoginControl control = new LoginControl(user, view);
    }
}
```

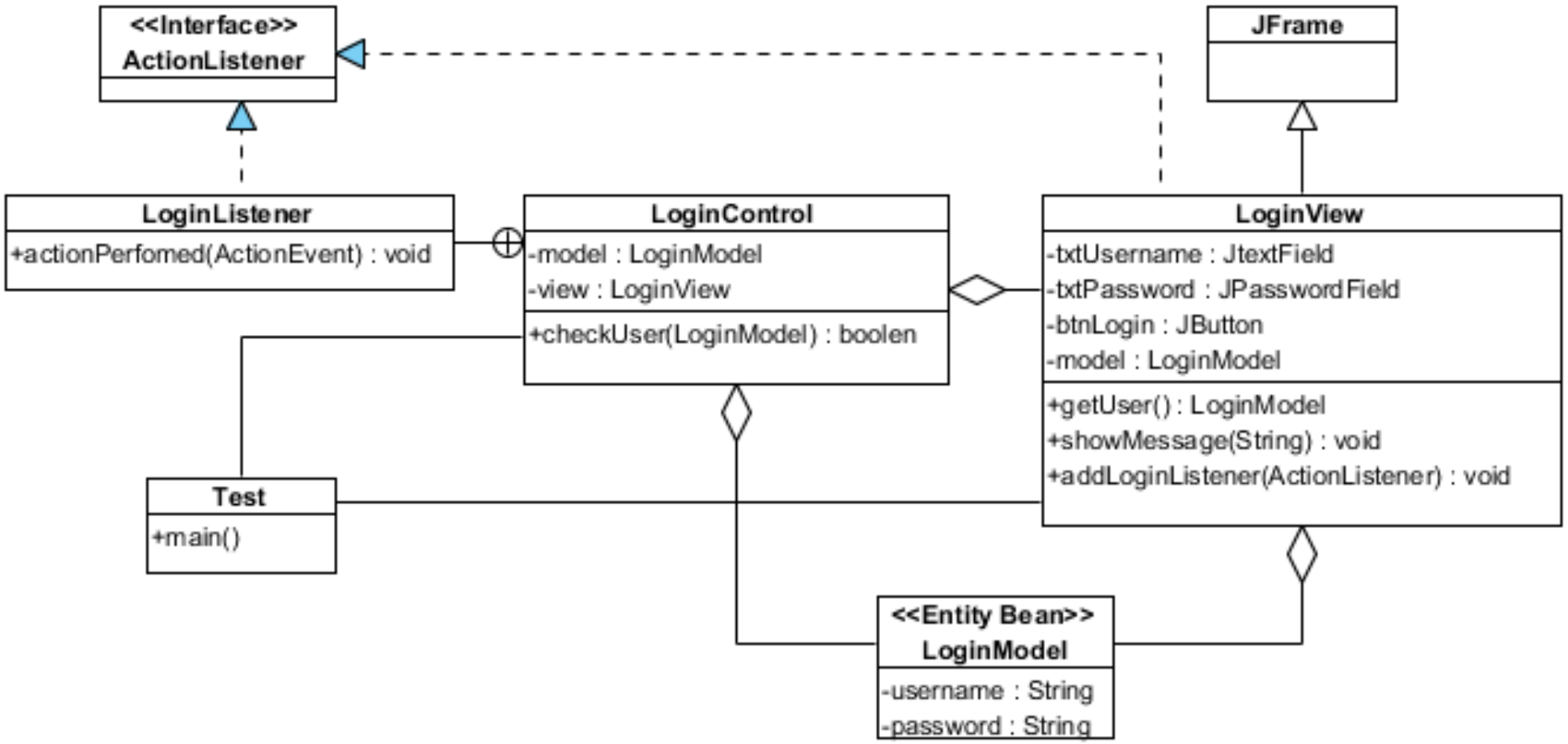
Case study: MVC với GUI

Yêu cầu bài toán

- Tạo một form đăng nhập gồm usename, password và một nút login
- Mỗi khi click vào nút login, chương trình phải kiểm tra thông tin đăng nhập có đúng không, nếu đúng thông báo thành công,
- nếu sai thông báo đăng nhập sai!

Xây dựng chương trình theo mô hình MVC

Sơ đồ các lớp


<<Interface>>
ActionListener
JFrame
LoginListener
+actionPerfomed(ActionEvent) : void
LoginControl
-model : LoginModel
-view : LoginView
+checkUser(LoginModel) : boolen
LoginView
-txtUsername : JtextField
-txtPassword : JPasswordField
-btnLogin : JButton
-model : LoginModel
+getUser() : LoginModel
+showMessage(String) : void
+addLoginListener(ActionListener) : void
Test
+main()
<<Entity Bean>>
LoginModel
-username : String
-password : String

LoginModel

```
public class LoginModel {
    private String userName;
    private String password;

    public LoginModel(){

    }

    public LoginModel(String username, String password){
        this.userName = username;
        this.password = password;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }
}
```

LoginView (1)

```
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;

public class LoginView extends JFrame implements ActionListener{
    private JTextField txtUsername;
    private JPasswordField txtPassword;
    private JButton btnLogin;
    private LoginModel model;
```

LoginView (2)

```
public LoginView(){
    super("Login MVC");

    txtUsername = new JTextField(15);
    txtPassword = new JPasswordField(15);
    txtPassword.setEchoChar('*');
    btnLogin = new JButton("Login");

    JPanel content = new JPanel();
    content.setLayout(new FlowLayout());
    content.add(new JLabel("Username:"));
    content.add(txtUsername);
    content.add(new JLabel("Password:"));
    content.add(txtPassword);
    content.add(btnLogin);


    this.setContentPane(content);
    this.pack();

    this.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            System.exit(0);
        }
    });
}
```

LoginView (3)

```
    public void actionPerformed(ActionEvent e) {
    }

    public LoginModel getUser(){
        model = new LoginModel(txtUsername.getText(),
                                        txtPassword.getText());
        return model;
    }

    public void showMessage(String msg){
        JOptionPane.showMessageDialog(this, msg);
    }

    public void addLoginListener(ActionListener log) {
          btnLogin.addActionListener(log);
        }
}
```

LoginControl (1)

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;


public class LoginControl {
    private LoginModel model;
    private LoginView view;

    public LoginControl(LoginView view){
        this.view = view;

        view.addLoginListener(new LoginListener());
    }
```

LoginControl (2)

```
class LoginListener implements ActionListener {
  public void actionPerformed(ActionEvent e) {
    try {
       model = view.getUser();
       if(checkUser(model))
         view.showMessage("Login succesfully!");
       else
         view.showMessage("Invalid username and/or password!");
    }catch (Exception ex) {
         view.showMessage(ex.getStackTrace().toString());
     }
  }
}
```

LoginControl (3)

```
public boolean checkUserJDBC(LoginModel user) throws Exception {

    String dbUrl = "jdbc:mysql://your.database.domain/yourDBname";
    String dbClass = "com.mysql.jdbc.Driver";
    String query = "Select * FROM users WHERE username ='"
        + user.getUserName() + "' AND password ='"
        + user.getPassword() + "'";

    try {
        Class.forName(dbClass);
        Connection con = DriverManager.getConnection (dbUrl);
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);

        if (rs.next()) {
          return true;
        }

        con.close();
    }catch(Exception e) {
        throw e;
    }
    return false;
  }
}
```

LoginControl (4)

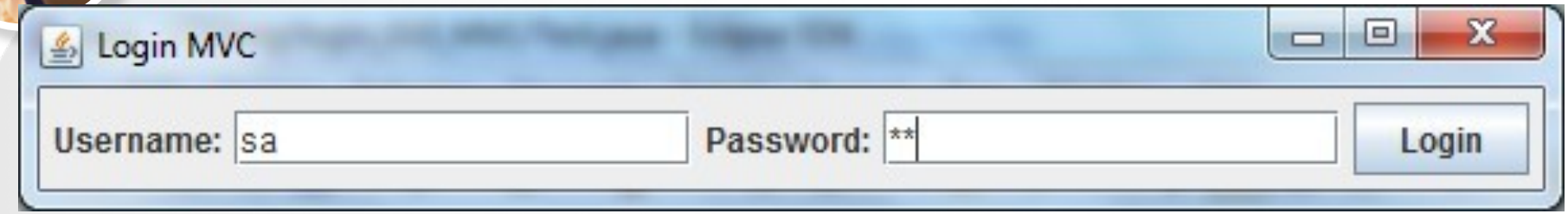
```
private boolean checkLogin(){
    if ((user.getUserName().equals("sa"))
        &&(user.getPassword().equals("sa") )){   return
            true;
    }
    return false;
}
}
```

Test

```
public class Test {
    public static void main(String[] args) {
        LoginView view       = new LoginView();
        LoginControl controller = new LoginControl(view);
        view.setVisible(true);
    }

}
```

Kết quả


Login MVC
Username:
sa
Password:
**
Login


Message
Login succesfully!
OK


Message
Invalid username and/or password!
OK

Questions?