

# CSCI 5448

## GROUP-23

# MEDIA PREFERENCE TRACKING (MPT)

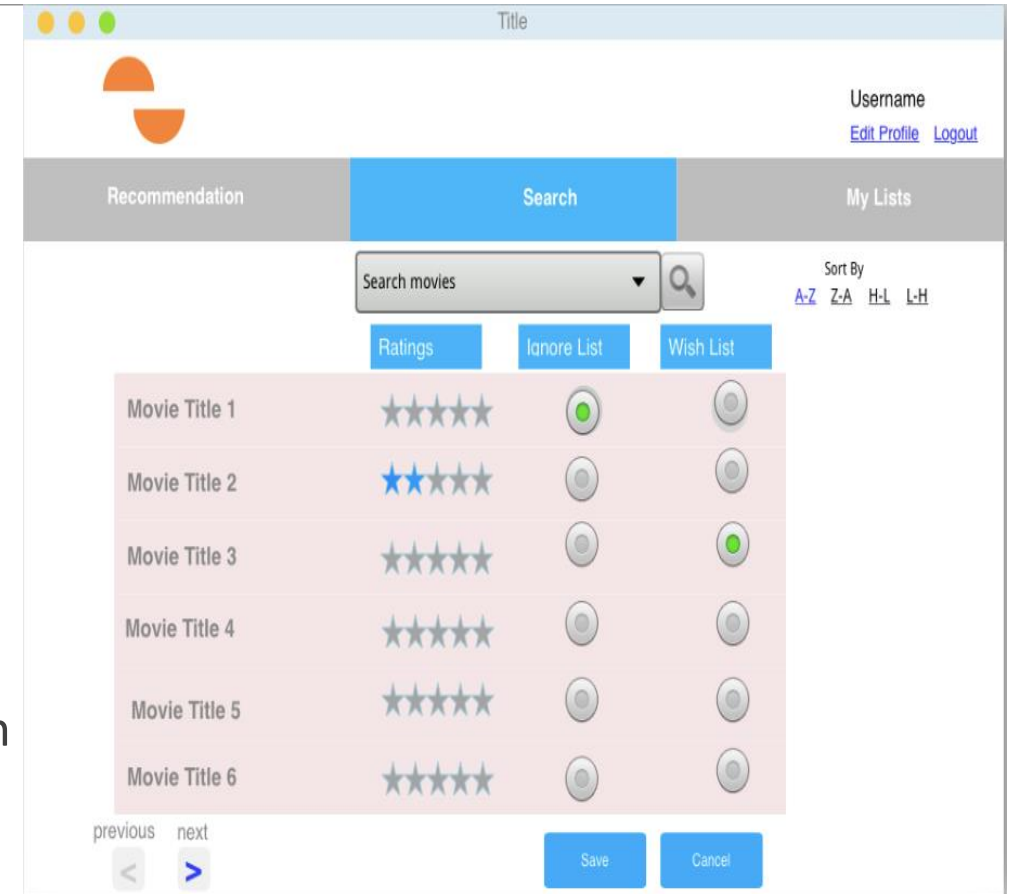
---

BY AVNEENDRA KANVA



# OVERVIEW: MPT

- Movie Preference Tracking application.
- User able to see all list of movies listed on the Search tab.
- User can do following:
  - i) Rate a movie
  - ii) Put the movie on his wish list.
  - iii) Put the movie on his ignore list.
- System recommends the list of movies to user based on preference of other users which matches them.



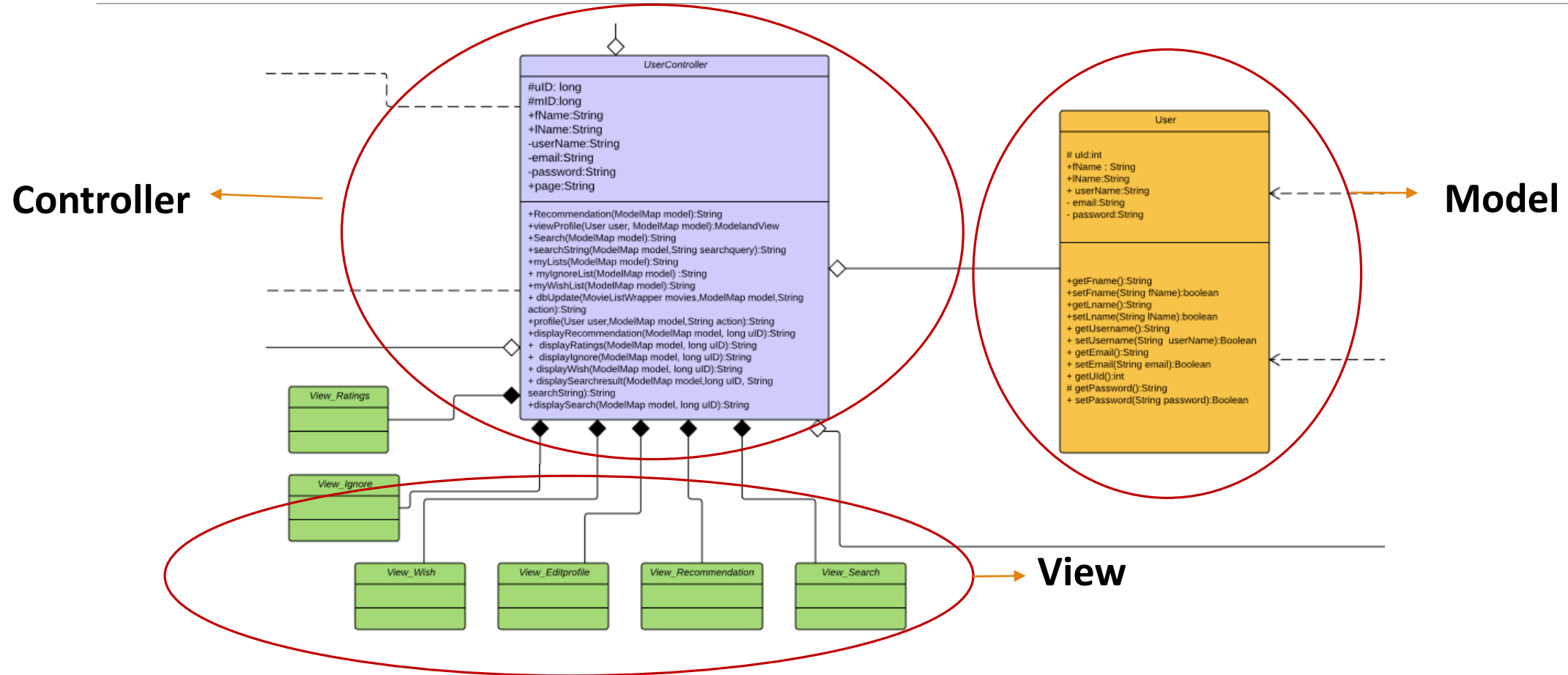
# SPRING MVC

---

- “The servlet way” works well for small web based applications. Difficult to organize for big applications.
- Spring MVC offers better separation of concerns.
- Spring offers structured framework using dependency-injection to integrate different technologies together.
- Integrates well with ORM Framework like Hibernate



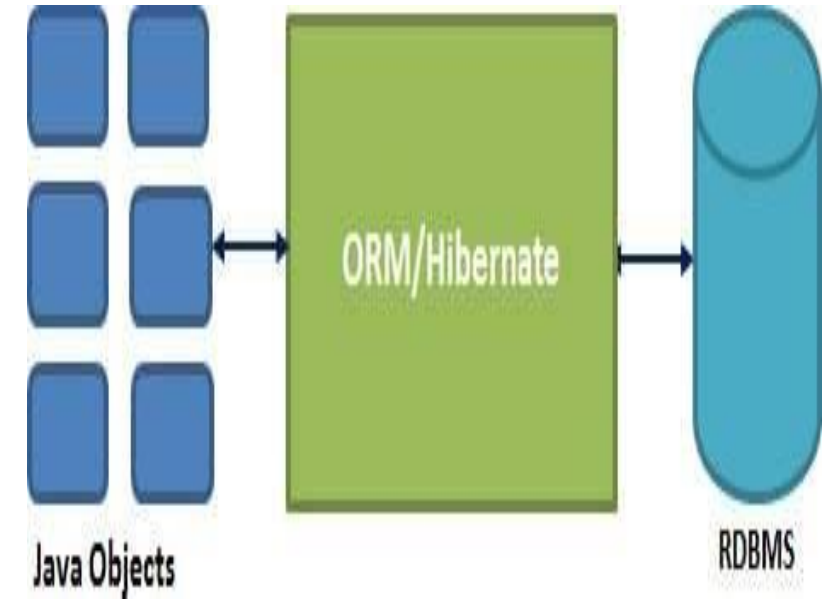
# MVC EXAMPLE



# HIBERNATE

---

- Easier to use than JDBC.
- Own query language which is database independent.
- It supports Inheritance, Association and Collection.
- It supports relationships like One-To-Many, Many-To-One and Many-To-Many.

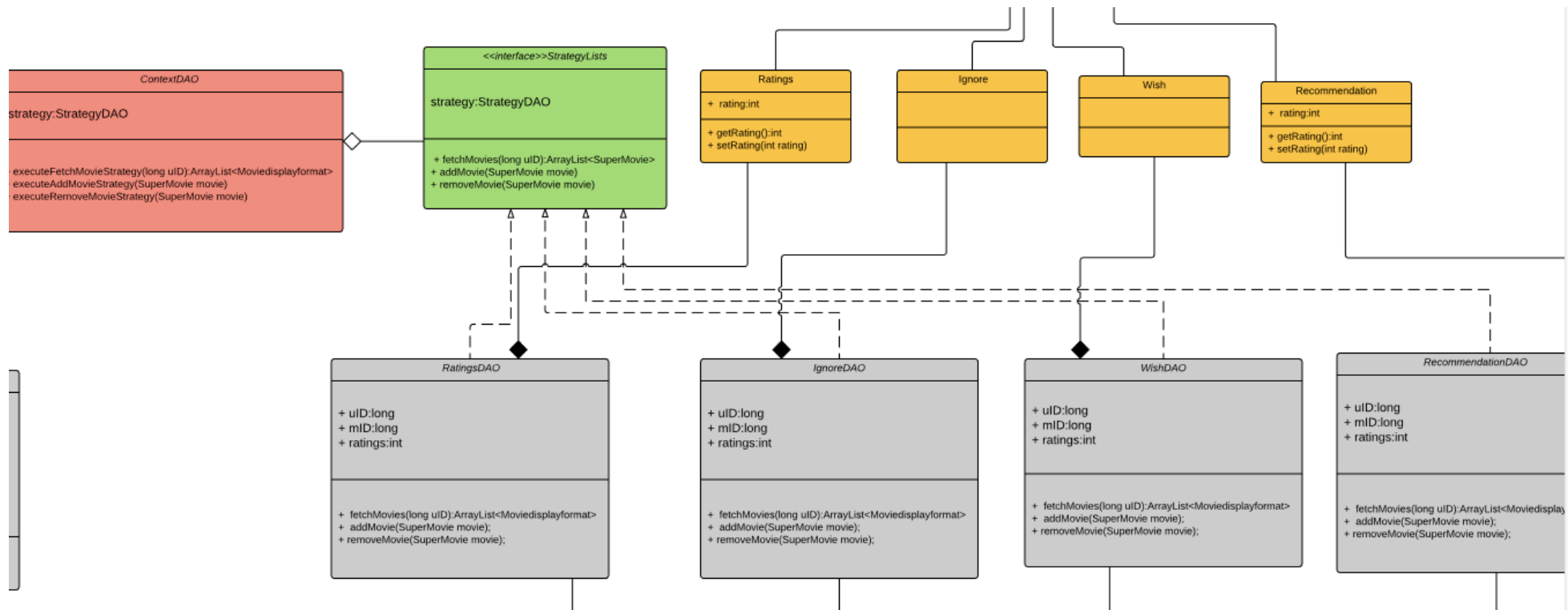


# Demo video

---

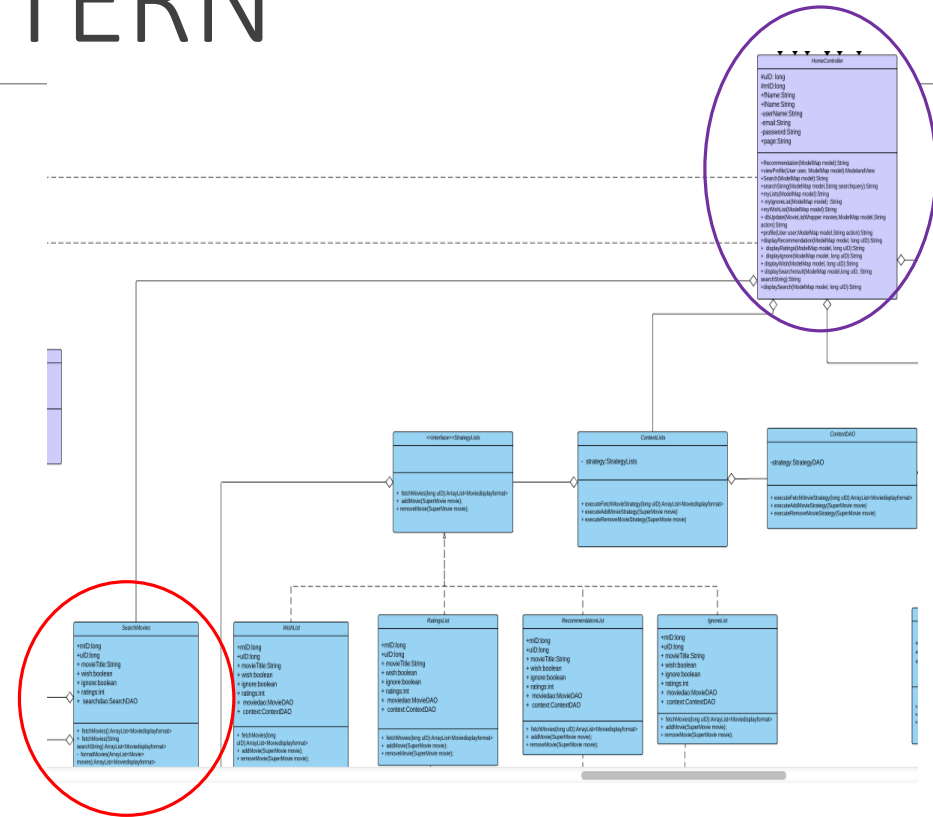
<http://screencast.com/t/sxLsG2MQ15r>

# STRATEGY DESIGN PATTERN



# ADAPTER DESIGN PATTERN

- For implementing Search List from movie database.
- It retrieves movies from database in different format than other lists.
- All the other lists such as ignore, wish etc. implement a common interface.
- Adapter could be used to integrate SearchMovies class within Strategy Design pattern.





# REFACTORING

---

- Used Extract method
- Used Shotgun Surgery.



# EXTRACT METHOD EXAMPLE

---

```
public ArrayList<Moviedisplayformat> fetchMovies()
{
    formatmovies=formatMovies(searchdao.fetchMovies());
    return formatmovies;
}

public ArrayList<Moviedisplayformat> fetchMovies(String searchString)
{
    formatmovies=formatMovies(searchdao.fetchMovies(searchString));
    return formatmovies;
}

public ArrayList<Moviedisplayformat> formatMovies(ArrayList<Movie> movies)
{
    for(Movie mov: movies)
    {
        movieTitle=mov.getMovieTitle();
        ratings=0;
        wish= false;
        ignore=false;

        Moviedisplayformat formatmovie = new Moviedisplayformat(movieTitle,wish,ignore,ratings);
        formatmovies.add(formatmovie);
    }
    return formatmovies;
}
```

# SHOTGUN SURGERY

---

```
public class SuperDAO
{
    ArrayList<SuperMovie> movies = new ArrayList<SuperMovie>();

    public Session beginSession()
    {
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        Session session = sessionFactory.openSession();
        session.beginTransaction();
        return session;
    }

    public ArrayList<SuperMovie> getResults(String queried, long uID)
    {
        Session session=beginSession();
        Query query = session.createQuery(queried);
        query.setParameter("uID",uID);

        @SuppressWarnings("unchecked")
        List<SuperMovie> allmovies = query.list();
        for(SuperMovie movie: allmovies)
        {
            movies.add(movie);
        }
        session.getTransaction().commit();
        return movies;
    }

    public void removeEntry(String query1,String query2,long mID)
    {
        Session session=beginSession();
        Query query = session.createQuery(query1);
        query.setParameter("mID",mID);
        if(query.uniqueResult()!=null)
        {
            query=session.createQuery(query2);
            query.setParameter("mID", mID);
        }
    }
}
```

# SHOTGUN SURGERY

---

```
public class WishDAO extends SuperDAO implements StrategyDAO
{
    public long uID,mID;
    public int ratings;
    Wish wsh = new Wish();
    @Override
    public ArrayList<SuperMovie> fetchMovies(long uID)
    {
        System.out.println("Inside fetchMovies in Wish DAO");
        ArrayList<SuperMovie> movies = new ArrayList<SuperMovie>();
        String queried="from Wish W where W.uID = :uID";

        movies=getResults(queried,uID);

        return movies;
    }
    public void addMovie(SuperMovie movie)
    {
        wsh= (Wish) movie;
        Session session=beginSession();
        session.saveOrUpdate(wsh);
        session.getTransaction().commit();
    }

    public void removeMovie(SuperMovie movie)
    {
        mID=movie.getMID();

        String query1="from Wish W where W.mID = :mID";
        String query2="delete from Wish W where W.mID = :mID";
        removeEntry(query1,query2,mID);
    }
}
```

---

THANK YOU!

A solid orange horizontal bar spanning the width of the slide at the bottom.