Works Done:

* Research Spring MVC Framework
  + DispatcherServlet class (front-controller): tất cả các request đều phải thông qua class này.
  + MVC: all the incoming request >>> DispatcherServlet >>> Hander Mapping (xml file) : kiểm tra các request và hướng đến controller tương ứng >>> Controller: handle requests and return an object of ModelAndView >>> Dispatcher: check view resolver in xml file and show return to View (jsp,…).
  + @Controller: this annotation marks a class as Controller.
  + @RequestMapping: this annotation map a request from client to a method ( or class ) tương ứng.
  + Web.xml: config all requests will be forwarded to the DispatcherServlet.
  + Define the bean in xml file to work with DispatcherServlet:
    - <context:component-scan>: where DispatcherServlet will search controller class.
    - Bean class InternalResourceViewResolver: to work with jsp page.
* Research Spring MVC Tiles: chia layout thành nhiều phần riêng biệt, dễ dàng cập nhật từng phần riêng biệt.
* Spring JDBC Templete: provides methods to write the queries directly, save lots of work and time.
  + JDBC – Java DataBase Connectivity
  + Method of Spring JDBC Template:
    - Public int update(String sql) : insert,update, delete records.
    - Public int update(String sql, Object…args): insert, update, delete using PreparedStatement given arguments.
    - Public void execute(String query) : to execute query.
    - Public T execute(Spring sql, PreparedStatementCallback action): execute query by PreparedStatement callback.
    - Public T query(String sql, ResultSetExtractor rse): fetch records using ResultsSetExtractor.
    - Public List query(String sql, RowMapper rse): fetch records using RowMapper.
  + PreparedStatement: to use execute() method with parameters query
    - Public T execute(String sql, PreparedStatementCallback<T>){\*};
    - PreparedStatementCallback: processes the input parameterized and output results.
    - Only one method:
      * \*Public T doInPreparedStatement(PreparedStatement ps) throws SQLException, DataAccessException{}
  + RowMapper:
    - Public T query(String sql, RowMapper<T> rm){\*}
    - \*Public T mapRow(ResultSet rs, int rowNumber) throws SQLException{}
  + SimpleJDBCTemplete
* Research Hibernate:
  + Inheritance Mapping:
    - Table Per Hierarchy: map the whole hierarchy by single table only. (by xml or annotation)
    - Table Per Concrete Class: map table with table having no relations to each other. (by xml or annotation)
    - Table Per Subclass: subclass mapped tables are related to parent class mapped table by primary key and foreign key relationship.
* Research Virtual-Host Apache Tomcat.
  + Create virtual host on local & virtual machine with Tomcat.
  + Configure file hosts: hostname with ipaddress from windows/system32/driver/etc/hosts(local) & /etc/hosts(virtual machine)
  + Configure server.xml in Tomcat: to create virtual host, create different hostname with the same ipaddress.
* RESTeasy:
  + REST: Representational State Transfer
  + Annotation Support In REST
    - @Path: path for a class or method.
    - @GET: get (select) resource from server.
    - @POST: create new resource
    - @DELETE: delete resource
    - @PUT: update resource.
    - @Produces: It specifies the MIME media type of the response of a resource.
    - @Consumes: It specifies the MIME media type of the request to a resource.
    - @PathParam: to extract a parameter from the uri path of resource.
    - @QueryParam: to extract a parameter form the uri path of resource.