Final Project Report

Silu Zhao; 001224750

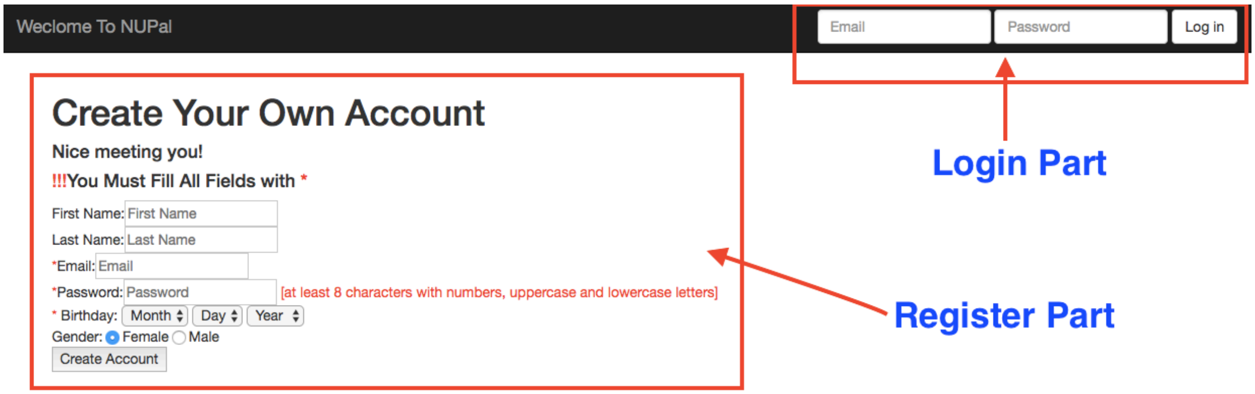
SUMMARY: It is the extension of my Web Design Project “NUPal Go!”. It is a blog platform for users to share their stories. There are two roles: Users and Admin. For customers, they can register using their email address. Then they can publish articles and follow friends to see what they have published. As for the admin, admin can manage the platform to make sure that they can control the purity of the platform. They can see the customer information and delete the customer. They can print the customer information in pdf format file it.

FUNCTIONALITY:

Users:

1> REGISTER – Register Controller
Email, Password and Birthday must be filled
Not duplication register with same pass and email
Invalid register will return to the home page

2> Login -- LoginController
Empty or Invalid email and pass pair will lead to the home page



Weclome To NUPal

Email Password Log in

Create Your Own Account

Nice meeting you!

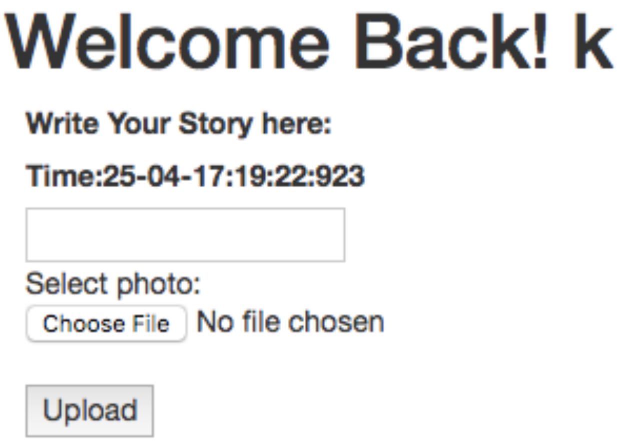
!!!You Must Fill All Fields with *

First Name: First Name
Last Name: Last Name
*Email: Email
*Password: Password [at least 8 characters with numbers, uppercase and lowercase letters]
* Birthday: Month Day Year
Gender: ● Female ○ Male
Create Account

Login Part

Register Part

3> Publish Articles -- FileUploadController
Customers can write down the articles and append the picture they want to publish their articles. Upload Picture is required while content is optional. If customers do not upload a picture. It is not permitted to save the article in database. If customers add a picture and hit the submit button. It will direct you to the successful published page to give a hint to customers.



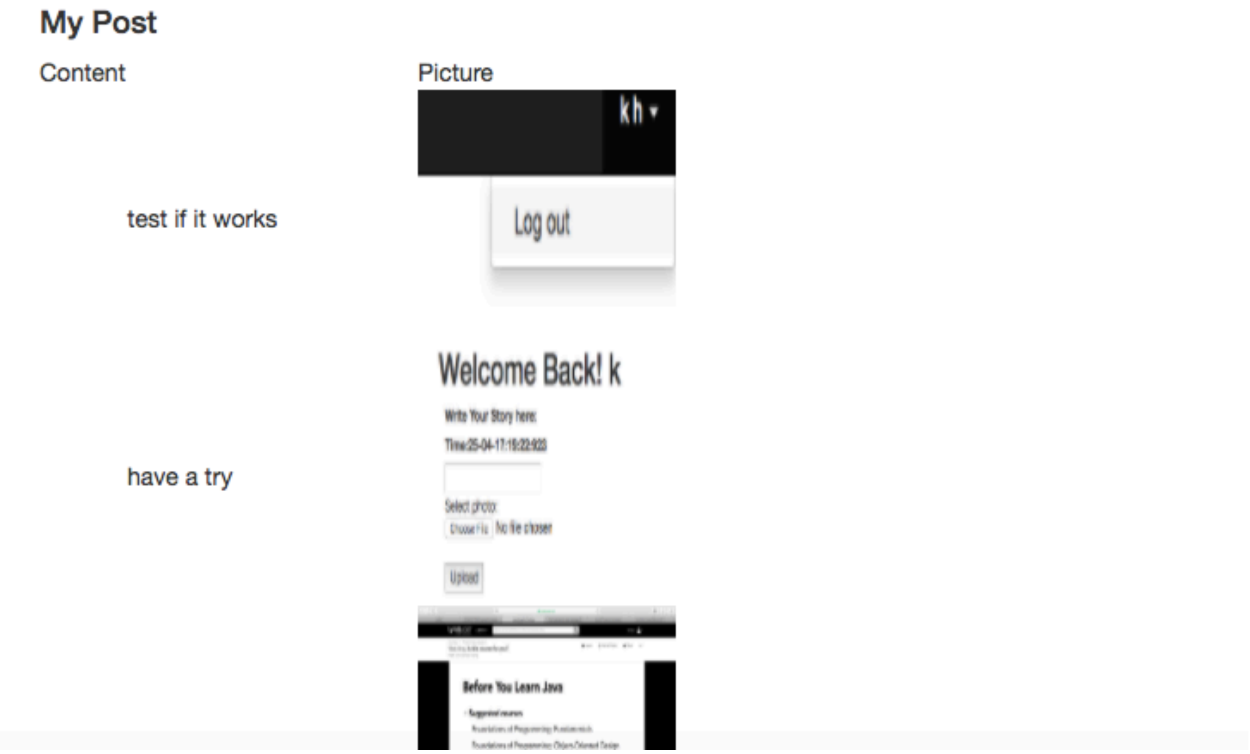
Welcome Back! k

Write Your Story here:

Time:25-04-17:19:22:923

Select photo:

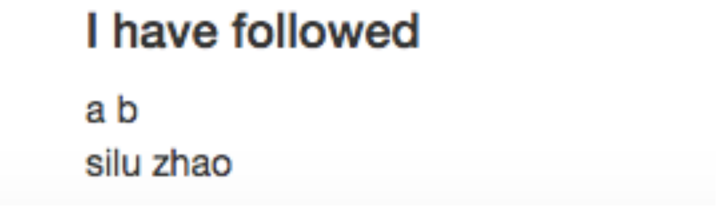
4> View Your Own Articles -- LoginController
Articles published by customer themselves will be listed in the home page after user login



My Post

Content	Picture
test if it works	
have a try	

5> View Friend list -- LoginController
Friend List follows the article list in the home page.

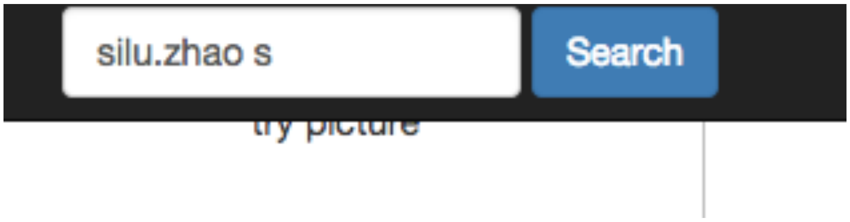


I have followed

a b

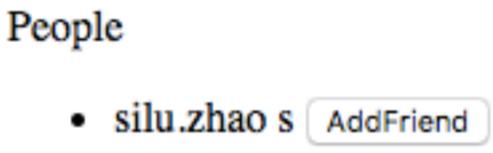
silu zhao

6> Add Friend – FindFriendsController & AddFriendsController
Users can search friends by their Name, including first name, last name and full name. Users will get a full list of names containing the keywords users input. User himself and friends that users already added to their friend list will not show.
Then user can adds the friends by clicking the add button.



silu.zhao s

[picture - search function]

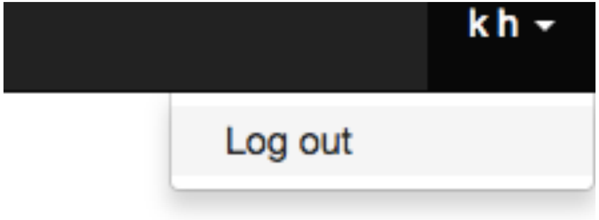


People

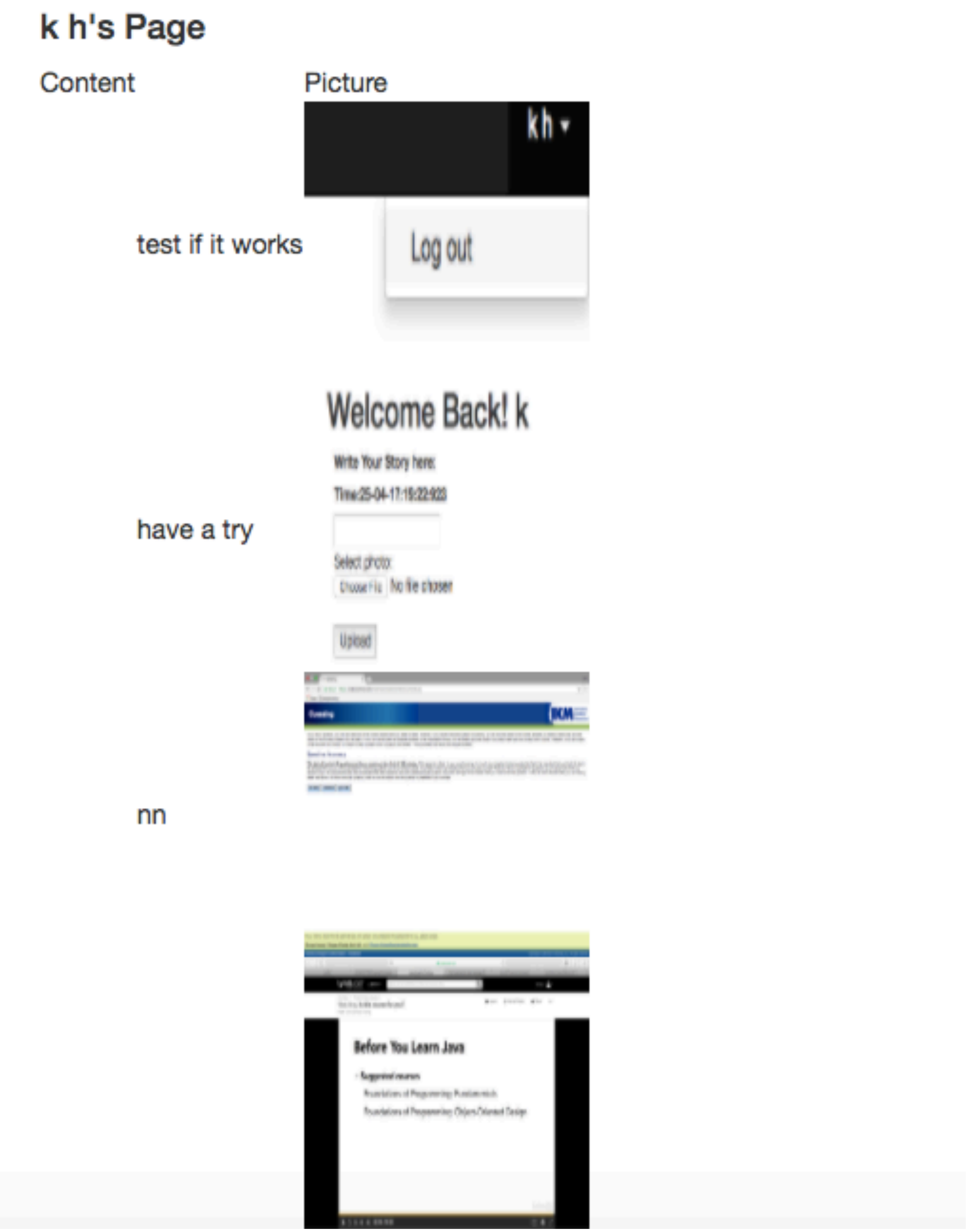
- silu.zhao s

[picture – search result]

7> Log Out -- FileUploadController
Click Log Out to return to the home page



8> View Friends Articles -- ShowFriendsArticleController
Users can click friend's name and go to friend's content page to view their published articles.



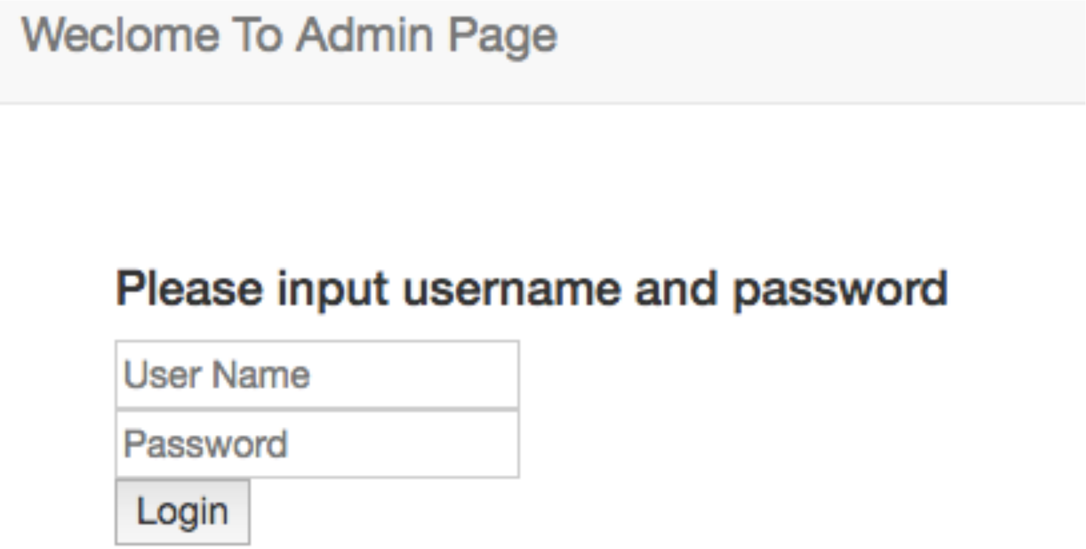
k h's Page

Content	Picture
test if it works	kh ▾ Log out
have a try	Welcome Back! k Write Your Story here: Time:25-04-17:19:22:923 Select photo: Choose File No file chosen Upload
nn	
	Before You Learn Java - Suggested courses Foundations of Programming: Fundamentals Foundations of Programming: Object-Oriented Design

Admin: -- AdminController

1> Login

Admin has to enter the right username and password pair to log in
The name and password has been written into the database.
Only the admin who is in charge of the database can add the admin authority to someone

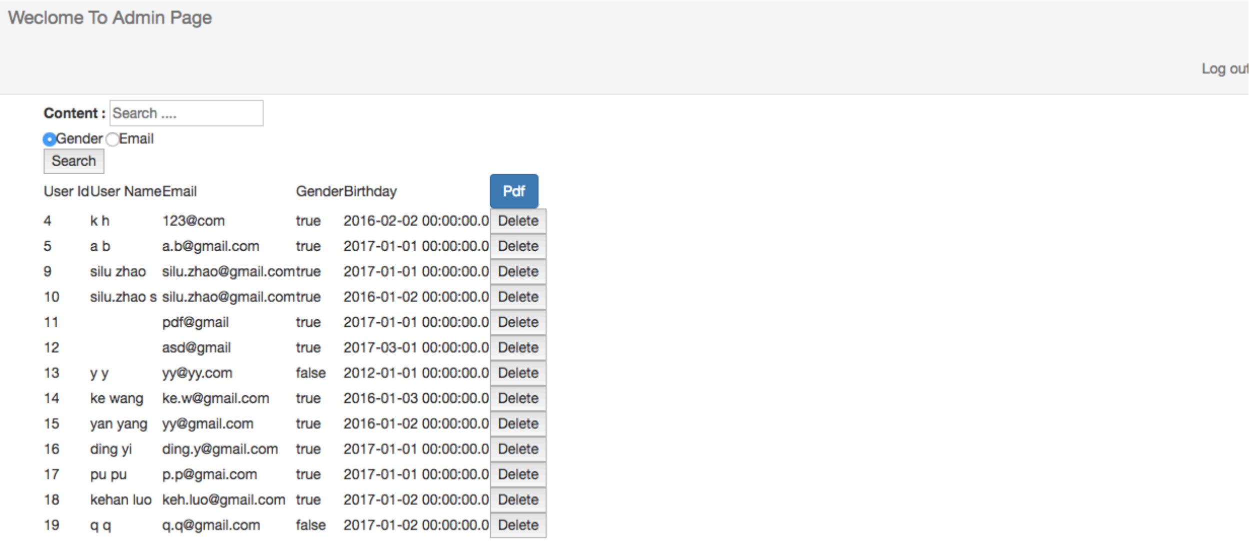


Weclome To Admin Page

Please input username and password

2> View User Information

After Admin login, they can see all users' information in the page



Weclome To Admin Page

Content : Search ....

● Gender ○ Email

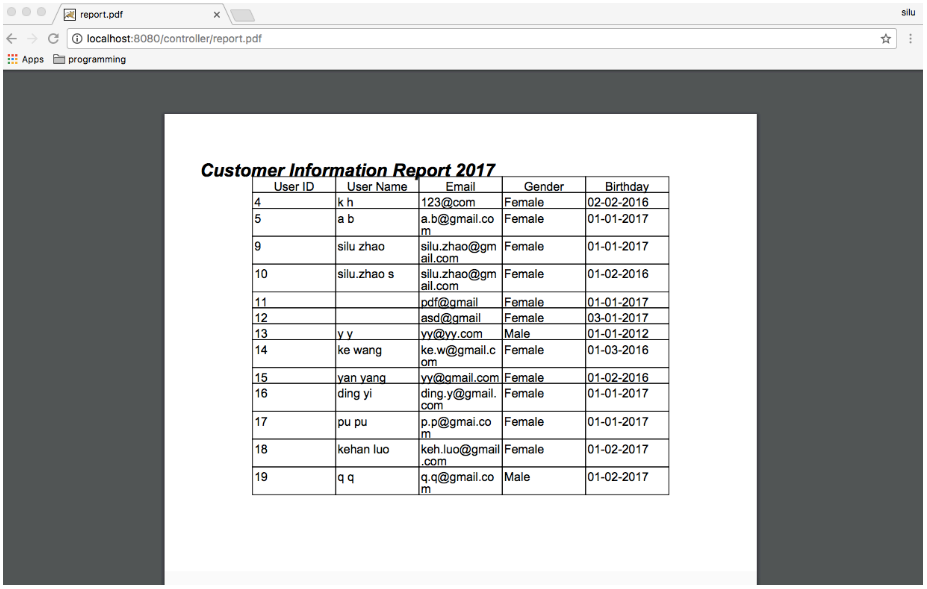
User Id	User Name	Email	Gender	Birthday	
4	k h	123@com	true	2016-02-02 00:00:00.0	Delete
5	a b	a.b@gmail.com	true	2017-01-01 00:00:00.0	Delete
9	silu zhao	silu.zhao@gmail.com	true	2017-01-01 00:00:00.0	Delete
10	silu.zhao s	silu.zhao@gmail.com	true	2016-01-02 00:00:00.0	Delete
11		pdf@gmail	true	2017-01-01 00:00:00.0	Delete
12		asd@gmail	true	2017-03-01 00:00:00.0	Delete
13	y y	yy@yy.com	false	2012-01-01 00:00:00.0	Delete
14	ke wang	ke.w@gmail.com	true	2016-01-03 00:00:00.0	Delete
15	yan yang	yy@gmail.com	true	2016-01-02 00:00:00.0	Delete
16	ding yi	ding.y@gmail.com	true	2017-01-01 00:00:00.0	Delete
17	pu pu	p.p@gmai.com	true	2017-01-01 00:00:00.0	Delete
18	kehan luo	keh.luo@gmail.com	true	2017-01-02 00:00:00.0	Delete
19	q q	q.q@gmail.com	false	2017-01-02 00:00:00.0	Delete

[Admin Page]

3> Delete Users

Click the delete button to delete users. After uses are deleted, user will not be able to login again.

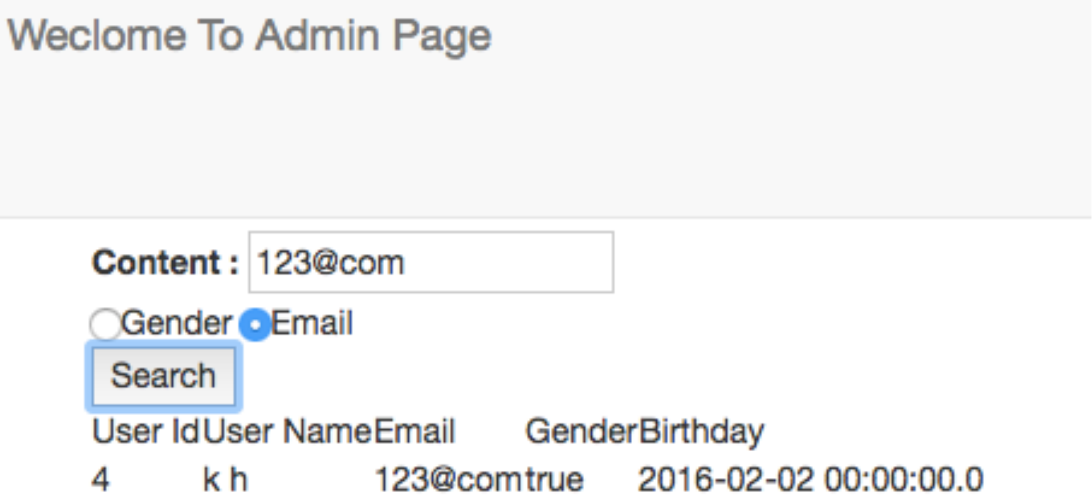
4> Print User Information as pdf format to file
They can use the pdf button to get pdf view to make admin print the useful information easier.



Customer Information Report 2017

User ID	User Name	Email	Gender	Birthday
4	k h	123@com	Female	02-02-2016
5	a b	a.b@gmail.com	Female	01-01-2017
9	silu zhao	silu.zhao@gmail.com	Female	01-01-2017
10	silu.zhao s	silu.zhao@gmail.com	Female	01-02-2016
11		pdf@gmail	Female	01-01-2017
12		asd@gmail	Female	03-01-2017
13	y y	yy@yy.com	Male	01-01-2012
14	ke wang	ke.w@gmail.com	Female	01-03-2016
15	yan yang	yy@gmail.com	Female	01-02-2016
16	ding yi	ding.y@gmail.com	Female	01-01-2017
17	pu pu	p.p@gmai.com	Female	01-01-2017
18	kehan luo	keh.luo@gmail.com	Female	01-02-2017
19	q q	q.q@gmail.com	Male	01-02-2017

5> Delete Users
Admin can use the search field to search the specific information they want by email or gender.



Weclome To Admin Page

Content : 123@com

○ Gender ● Email

User Id	User Name	Email	Gender	Birthday
4	k h	123@com	true	2016-02-02 00:00:00.0

[Search Result]

Appendix:

1> AddFriendController

```
package com.nupal.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.nupal.dao.PersonDAO;
@Controller
public class AddFriendController {

    @Autowired
    @Qualifier("personDAO")
    PersonDAO personDAO;

    @RequestMapping(value="/addfriends*", method = RequestMethod.GET)
    public ModelAndView findFriends(HttpServletRequest request){

        long cusid = Long.parseLong(request.getParameter("cusid"));
        long fid = Long.parseLong(request.getParameter("friendid"));
        personDAO.addFriends(cusid,fid);

        return null;
    }

}
```

2> AdminController

```
package com.nupal.controller;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

import com.nupal.dao.PersonDAO;
import com.nupal.exception.PersonException;
import com.nupal.pojo.Admin;
import com.nupal.pojo.Customer;
import com.nupal.pojo.Person;

@Controller
public class AdminController {

    @Autowired
    @Qualifier("personDAO")
    PersonDAO personDAO;

    @RequestMapping(value = "/login-admin.htm", method = RequestMethod.GET)
    public ModelAndView adminPage(HttpServletRequest request) {
        //HttpSession session = request.getSession();

        ModelAndView model = new ModelAndView();

        try{
            List<Customer> personList = personDAO.listUser();
            //Person p = personDAO.registerAdmin();
            model.setViewName("admin-login");
            model.addObject("userList",personList);

            //session.setAttribute("userList", personList);
            return model;
        }catch (PersonException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return null;
    }

    @RequestMapping(value = "/login-admin.htm", method = RequestMethod.POST)
    public ModelAndView ShowForm(HttpServletRequest request){
            HttpSession session = request.getSession();
            ModelAndView model = new ModelAndView();

            String userName = request.getParameter("username");
String password = request.getParameter("password");
Admin admin = null;
try {
    admin = personDAO.findAdmin(userName, password);
    session.setAttribute("user", admin);
    List<Customer> personList = personDAO.listUser();
                    model.setViewName("admin-form");
                    model.addObject("userList",personList);

                    session.setAttribute("userList", personList);

                    return model;
            } catch (PersonException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }

if(admin != null){
    return new ModelAndView("admin-login");
}else{
    return new ModelAndView("error");
}
    }

    @RequestMapping(value = "/admin-deleteUser*", method = RequestMethod.GET)
    public void deleteUser(HttpServletRequest request){
            long uid = Integer.parseInt(request.getParameter("cusid"));

            try {
                    List<Customer> personList = personDAO.listUser();

                    for(Customer per:personList){
                            if(per.getPersonID() == uid){
                                    personDAO.delete(per);
                            }
                    }
            } catch (PersonException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
    }

    @RequestMapping(value = "/admin-search*", method = RequestMethod.GET)
    @ResponseBody
    public String searchUser(HttpServletRequest request){

        List<Customer> customers = new ArrayList<Customer>();
        String keywords = request.getParameter("keywords");
        String option = request.getParameter("option");
        try {
            List<Customer> personList = personDAO.listUser();
            //Iterator<Customer> itr = personList.iterator();
            boolean g=true;
            if(option.equals("gender")){

                if(keywords.equalsIgnoreCase("male")){
                    for(Customer per:personList){
                            if(!per.isGender())
                                customers.add(per);
                    }
                    //g=false;
                }else if(keywords.equalsIgnoreCase("female")){
                    for(Customer per:personList){
                            if(per.isGender())
                                customers.add(per);
                    }
                    //g=false;
                }else{
                    for(Customer per:personList)
                            customers.add(per);
                }

                /*
                for(Customer per:personList){
                    if(g){
                        if(per.isGender())
                        customers.add(per);
                    }else{
                        if(!per.isGender())
                            customers.add(per);
                    }
                }*/
            }
            if(option.equals("email")){
                for(Customer per:personList){
                    if(per.getEmail().equalsIgnoreCase(keywords)){
                        customers.add(per);
                    }
                }

                if(keywords.isEmpty()){
                    for(Customer per:personList){
                            customers.add(per);
                    }
                }
            }

            String rsp = "<table>"+
          "<tr>"+
      "<td>User Id</td>"+
      "<td>User Name</td>"+
      "<td>Email</td>"+
      "<td>Gender</td>"+
      "<td>Birthday</td>"+
      "</tr>";

            for(Customer cus:customers){
                rsp = rsp + "<tr><td>"+ cus.getPersonID()+"</td>"+
                            "<td>"+cus.getFname()+"
"+cus.getLname()+"</td>"+
                            "<td>"+cus.getEmail()+"</td>"+
                            "<td>"+cus.isGender()+"</td>"+
                            "<td>"+cus.getBirth()+"</td>"+
                            "</tr>";
            }
            rsp = rsp+"</table>";

            return rsp;

        } catch (PersonException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return null;
    }

    @RequestMapping(value = "/adlogout.htm", method = RequestMethod.GET)
```

```
        public String showForm(HttpServletRequest request) {

                HttpSession session = request.getSession();
                session.invalidate();
                // return form view
                return "admin-login";

        }
}
```

3> FileUploadController

```
package com.nupal.controller;

import java.io.File;
import java.io.IOException;
import java.net.ContentHandler;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import javax.swing.text.AbstractDocument.Content;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.propertyeditors.CustomDateEditor;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.multipart.commons.CommonsMultipartFile;

//import com.me.dao.UserDAO;
//import com.me.upload.Qualifier;
//import com.me.upload.Qualifier;
//import com.me.validator.UserValidator;
import com.nupal.dao.ArticleDAO;
import com.nupal.dao.PersonDAO;
import com.nupal.exception.ArticleException;
import com.nupal.pojo.Article;
import com.nupal.pojo.Customer;
import com.nupal.validator.ArticleValidator;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class FileUploadController {

    @Autowired
    @Qualifier("articleDAO")
    ArticleDAO articleDAO;

    @Autowired
    @Qualifier("personDAO")
    PersonDAO personDAO;

    //ArticleDAO articleDAO = new ArticleDAO();
    //PersonDAO personDAS = new PersonDAO();

    //ArticleValidator articleValidator;

    @Autowired
    ServletContext servletContext;

    /*
    public void initBinder(WebDataBinder binder) {
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

        binder.registerCustomEditor(Date.class, new CustomDateEditor(dateFormat,
true));
    }*/

    @RequestMapping(value = "/upload.htm", method = RequestMethod.GET)
    public String showForm(ModelMap model) {
        Article article = new Article(); // should be AutoWired
        //article.setFavFramework(new String []{"Spring MVC"});

        // command object
        model.addAttribute("article", article);

        // return form view
        return "customer-form";
    }

    @RequestMapping(value = "/upload.htm", method = RequestMethod.POST)
    public String handleUpload(@ModelAttribute("article")  Article article,
HttpServletRequest request) {
        HttpSession session = request.getSession();
        Customer customer = (Customer)session.getAttribute("customer");
        String content = request.getParameter("article");
        String authorID =request.getParameter("author_id");
        //articleValidator.validate(article, content);
        /*
        if(content.equals(null)){
            return "publishFail";
        }*/

        try {

                File directory;
                String check = File.separator; // Checking if system is linux
                                                //
based or windows based by
                                                //
checking seprator used.
                article.setAuthor((Customer)session.getAttribute("customer"));

                String path = null;
                if (check.equalsIgnoreCase("\\")) {
                    path = servletContext.getRealPath("").replace("build\\",
""); // gives real path as Lab9/build/web/

                                // so we need to replace build in
the path

                                        }

                if (check.equalsIgnoreCase("/")) {
                    path = servletContext.getRealPath("").replace("build/",
"");

                    path += "/"; // Adding trailing slash for Mac systems.
                }
                directory = new File("/Users/silu.zhao/Desktop/STS/images");
                boolean temp = directory.exists();

                if (!temp) {
                    temp = directory.mkdir();
                }
                if (temp) {
                    // We need to transfer to a file

                    CommonsMultipartFile photoInMemory = 
article.getPhoto();
                    if(photoInMemory.isEmpty())
                    {
                        System.out.println("photoInMemory is null!!!!");
                    }
                    String fileName = 
photoInMemory.getOriginalFilename();
                    // could generate file names as well
//                  if(fileName == null){
//                      System.out.println("null");
//                  }
//                  System.out.println(fileName);
                    File localFile = new File(directory.getPath(), fileName);

                    // move the file from memory to the file

                    photoInMemory.transferTo(localFile);
                    article.setFilename(fileName);
                    System.out.println("File is stored at" + 
localFile.getPath());
                    System.out.print("registerNewUser");


    //article.setAuthor((Customer)session.getAttribute("customer"));

                    Article a = 
articleDAO.register(article,customer.getPersonID());

                } else {
                    System.out.println("Failed to create directory!");
                }



        } catch (IllegalStateException e) {
            System.out.println("*** IllegalStateException: " + e.getMessage());
        } catch (IOException e) {
            // TODO Auto-generated catch block
            System.out.println("*** IOException: " + e.getMessage());
        } //catch (ArticleException e) {
            // TODO Auto-generated catch block
            //e.printStackTrace();
        //}
 catch (ArticleException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return "publish-success";
    }

    @RequestMapping(value = "/logout.htm", method = RequestMethod.GET)
    public String showForm(HttpServletRequest request) {

        HttpSession session = request.getSession();
        session.invalidate();
        // return form view
        return "home";

    }

}
```

4> FindFriendsController

```
package com.nupal.controller;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Set;

//import javax.persistence.Query;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import com.nupal.dao.PersonDAO;
import com.nupal.exception.PersonException;
import com.nupal.pojo.Customer;
import com.nupal.pojo.Person;

import org.hibernate.Query;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class FindFriendsController {

        @Autowired
        @Qualifier("personDAO")
        PersonDAO personDAO;

        @RequestMapping(value="/findfriends.htm",method = RequestMethod.GET)
        public ModelAndView findFriends(HttpServletRequest request){
                HttpSession session = request.getSession();
                Customer customer = (Customer) session.getAttribute("customer");
                long cusID = customer.getPersonID();

                String requirement = request.getParameter("friendName");
                if(requirement.isEmpty()){
                        return new ModelAndView("find-friends-result","friendList",null);
                }
                String[] names = requirement.split("\\s+");

                List<Person> friendList = null;
                try {
                        friendList = personDAO.getFriends(names,cusID);
                        Set<Long> myFriends = customer.getFriendIDList();
                        myFriends.add(cusID);
                        //System.out.println("FRIEND SIZE"+friendList.size());
                         for (Iterator<Person> it = friendList.iterator(); it.hasNext();) {
                             if(myFriends.contains(it.next().getPersonID())){
                                 it.remove();
                             }
                           }

                        return new ModelAndView("find-friends-result","friendList",friendList);
                } catch (PersonException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                return null;
        }
}
```

5> HomeController

```
package com.nupal.controller;

import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 * Handles requests for the application home page.
 */
@Controller
public class HomeController {

        private static final Logger logger = LoggerFactory.getLogger(HomeController.class);

        /**
         * Simply selects the home view to render by returning its name.
         */
        @RequestMapping(value = "/", method = RequestMethod.GET)
        public String home(Locale locale, Model model) {
                logger.info("Welcome home! The client locale is {}.", locale);

                Date date = new Date();
                DateFormat dateFormat = 
DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG, locale);

                String formattedDate = dateFormat.format(date);

                model.addAttribute("serverTime", formattedDate );

                return "home";
        }
}
```

6> LoginController

```
package com.nupal.controller;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.nupal.dao.PersonDAO;
import com.nupal.exception.PersonException;
import com.nupal.pojo.Article;
import com.nupal.pojo.Customer;
import com.nupal.pojo.Person;

@Controller
public class LoginController {

        @Autowired
        @Qualifier("personDAO")
        PersonDAO personDao;

        @RequestMapping(value = "/login.htm", method = RequestMethod.POST)
        public ModelAndView loginMethod(HttpServletRequest request){
                HttpSession session = request.getSession();

                String email = request.getParameter("email");
     String password = request.getParameter("password");
     ModelAndView mv = new ModelAndView();

     //PersonDAO personDao = new PersonDAO();
```

```
    Customer customer = null;
    Article article = null;
    try {
                    customer = personDao.findCustomer(email, password);
                    if(customer ==null){
                            return new ModelAndView("home");
                    }
                    article = new Article();
                    List<Customer> friendList = new ArrayList<Customer>();

                    Date date = new Date();
                    String str = new SimpleDateFormat("dd-MM-
yy:HH:mm:SS").format(date);

                    friendList = personDao.listFriends(customer.getFriendIDList());
                    customer.setFriendList(friendList);
                    mv.addObject("customer", customer);
                    mv.addObject("article",article);
                    mv.addObject("flist",friendList);
                    mv.addObject("date",str);
                    mv.setViewName("customer-form");

                    session.setAttribute("customer", customer);
            } catch (PersonException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
    if(customer != null){
        return mv;
    }else{
        return new ModelAndView("error");
    }
        }
}
```

7> PdfReportController

```
package com.nupal.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.View;
import com.nupal.pdfView.PdfView;

@Controller
public class PdfReportController {

        @RequestMapping(value = "/report.pdf", method = RequestMethod.GET)
        public ModelAndView showPdfReport()
        {
                View view = new PdfView();
                return new ModelAndView(view);
        }

}
```

8> RegisterController

```
package com.nupal.controller;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.hibernate.Session;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.nupal.dao.PersonDAO;
import com.nupal.pojo.Article;
import com.nupal.pojo.Customer;
import com.nupal.pojo.Person.Role;
import com.nupal.validator.PersonValidator;
```

```
@Controller
public class RegisterController {

        @Autowired
        @Qualifier("personDAO")
        PersonDAO personDAO;

        @RequestMapping(value = "/add.htm", method = RequestMethod.POST)
        public ModelAndView register(HttpServletRequest request){

                HttpSession session = request.getSession();
                ModelAndView mv = new ModelAndView();

                Customer customer=null;
                try {
                // Person Property
                String fname = request.getParameter("fname");
     String lname = request.getParameter("lname");
     String email = request.getParameter("email");
     String password = request.getParameter("password");
     String gender = request.getParameter("gender");

     String birthday_year = request.getParameter("birthday_year");
     String birthday_month = request.getParameter("birthday_month");
     String birthday_day = request.getParameter("birthday_day");

     String birth = birthday_month + "-" + birthday_day +"-" + birthday_year;
     DateFormat format = new SimpleDateFormat("MM-dd-yyyy");

     if(email.isEmpty()||password.isEmpty()||birth.isEmpty()){
                        return new ModelAndView("home");
                }

     Date birthday = format.parse(birth);

     //save
                //PersonDAO personDao = new PersonDAO();
                Article article = new Article();
                customer = new Customer(Role.CUSTOMER);
                customer.setFname(fname);
                customer.setLname(lname);
                customer.setEmail(email);
                customer.setPassword(password);
                customer.setBirth(birthday);
                session.setAttribute("customer", customer);

                mv.addObject("customer", customer);
                mv.addObject("article",article);
                mv.setViewName("customer-form");

                if(gender.equals("female"))
                        customer.setGender(true);
                else
                        customer.setGender(false);

                Customer c = personDAO.findCustomer(email, password);
                if(c != null){
                        //personDAO.register(customer);
                        return new ModelAndView("home");
                }

                personDAO.register(customer);
                }catch (ParseException e) {
               e.printStackTrace();
                }catch(Exception e){
                        System.out.println("*** IllegalStateException: ");
                }
                //return new ModelAndView("home");
                return mv;
        }
}
```

9> ShowFriendArticleController

```
package com.nupal.controller;

import java.util.HashSet;
import java.util.List;
import java.util.Set;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.nupal.dao.ArticleDAO;
import com.nupal.dao.PersonDAO;
import com.nupal.exception.PersonException;
import com.nupal.pojo.Article;
import com.nupal.pojo.Customer;

@Controller
public class ShowFriendArticleController {


    @Autowired
    @Qualifier("articleDAO")
    ArticleDAO articleDAO;

    @Autowired
    @Qualifier("personDAO")
    PersonDAO personDAO;

    @RequestMapping(value = "/friendsArticle*", method = RequestMethod.GET)
    public ModelAndView showForm(HttpServletRequest request) {

        ModelAndView mv = new ModelAndView();
        long cusId = Long.parseLong(request.getParameter("fid"));
        Set<Long> fids = new HashSet<Long>();
        fids.add( cusId);
        try {
            List<Customer> friend = personDAO.listFriends(fids);
            mv.addObject("friend",friend.get(0));


        } catch (PersonException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        mv.setViewName("friendsArticle");
        return mv;

    }

}
```