



CS 319 - Object-Oriented Software Engineering
Final Report

v.map

Group 3-C

Yağız Gani

Kanan Asadov

Ömer Faruk Karakaya

Mert Osman Dönmezyürek

Table Of Contents

1. Introduction	2
2. User Guide	2
3. Changes and Additions	3
3.1 Session Management	3
3.2 Model	4
3.3 E-Mail Service	5
3.4 Views	5
4. Difficulties	7
4.1. Session Management	7
5. Incompleted parts	8
5.1 Custom Images for Announcements	8
6. References	8

1. Introduction

This final report includes the user guide on how to install the program, also the changes, additions and difficulties during the implementation process.

2. User Guide

Since v.map is a web application, it does not require any extra installation but a javascript supported web browser on any operating system on the user's side, which are the followings:

Operating Systems:

Windows, Mac, Linux / Ubuntu, Google Chrome OS (Chromebook)

Web Browsers:

Google Chrome v34 or later, Mozilla Firefox v34 or later, Internet Explorer v10 or later, all versions of Microsoft Edge or Apple Safari v6 or later

For the server side, system requires following applications to be installed on the server:

JDK 8+ and JRE 8+

Java Application Server (JBoss Wildfly 10)

PostgreSQL 9.X

All of them are compatible with Linux, Windows or MacOS distributions. To run the web application, the source files should be uploaded on the server.

3. Changes and Additions

3.1 Session Management

*SessionManager class is removed. The function of this class is handled SessionView class.

SessionUtils :

SessionUtils class is added inside Session package. This class basically controls the user utilities. Also some methods are added inside this class. These are:

```
public static HttpSession getSession(){
    return (HttpSession) FacesContext.getCurrentInstance()
        .getExternalContext().getSession(false);
} gets the current session.
public static HttpServletRequest getRequest(): It requests for HttpServlet access.
public static String getUsername(): It gets the username from the session object created inside
the function. Session object belongs to HttpSession class.
public static String getUserId(): "HttpSession session" object is created inside the function and if
this object is non-empty, the function returns "session.getAttribute("userId"); "
```

3.2 Model

UserManager class:

UserManager in the Session package is moved to Model package.

The functions and methods in UserManager class are modified:

"private List<User> users;" is added. //An arraylist of User objects.

"private User user;" is removed.

Constructor is added which basically creates the arraylist of User objects, dynamically.

public static void createUser(String fullName, String email, String password, boolean isAdmin) is added.

public boolean createAnnouncement(Announcement announcement) is deleted. No need to implement it in the UserManager, anymore.

public boolean deleteAnnouncement(int announcementID) is deleted. There is no need to implement the method on UserManager class.

public boolean changePassword() is deleted because there is no need to implement here on UserManager class.

setUser and getUser methods are also deleted. No need to use them anymore due to the some modifications on createUser() method.

public boolean editAnnouncement(Announcement announcement) is also deleted. Because, it is not needed anymore.

public boolean logout() function is removed.

3.3 E-Mail Service

EmailService Class:

private string prepareHTMLEmail(String text), this method was planned to create an email template in HTML by inserting the given text into the HTML codes and return string of the codes. However, it doesn't create a user-friendly email template. Therefore, it is decided to change it as **private String**

prepareVerificationEmailInHTML(String userFullName, String verificationCode) method. Instead of generating a general template for all emails, this method creates a specific well-designed verification e-mail template in HTML by taking the full name of the user and the verification code, then returns HTML code in string.

private void sendEmail (String recieverEmail, String mailSubject, String contentOfEmail), this method was meant to be a void method but other classes that use EmailService class need a feedback on whether the e-mail is sent successfully or not. Hence, the type of the class has been changed as boolean type to notify its user by returning true if the process is successful or false if it is failed.

3.4 Views

SessionView:

On views package, SessionManager class is modified:

public boolean isUser() checks if the person is whether admin or user. If the session object is not null, then the function returns **isAdmin()**. Otherwise the function returns false.

public boolean isAdmin() checks if the person is user or Admin. If the session object != null; then **return session.getAttribute("isAdmin").equals("true")**.

Else, the function returns false.

public boolean isGuest() is added. Returns true if the session object is equal to null.

`public boolean logIn()`: user object is filled by calling `getUser` method of `DatabaseManager` class.

`public void logOut()`: `session.invalidate()` is done.

`public void setPassword(String password)`: Sets the local password to the global password.

`public String getPassword()`: Returns password.

`public String setEmail()`: Sets the email String to the global email String

`public String getEmail()`: Returns email String.

Also try-catch statement is used so that, if the username or password is incorrect then the system gives an error message saying that: **"Incorrect Username and Password, Please enter correct username and Password"**.

*`public boolean isAdmin()` method is added to the class using the User class object. It basically checks whether the user is admin or not, by accessing the User class' `isAdmin()` method.

CreateAnnouncementView:

Functionality of this class is previously designed as a part of `menuView` but later separated as a new class for the sake of modularity. Most Important function is:

`public void onGeocode(GeocodeEvent event)` to create coordinates over textual adress with GoogleMaps JavaScript API

This class is make background works when user is adding new Announcement.

ManageEntriesView.java:

Functionality of this class is previously designed as a part of `menuView` but later separated as a new class for the sake of modularity. This class makes backgroud works when user deletes his/her previous announcements.

ManageUsersView.java:

Functionality of this class is previously designed as a part of `menuView` but later separated as a new class for the sake of modularity. This class makes backgroud works when admin deletes other users.

4. Difficulties

4.1. Session Management

The implementation of `public boolean signIn(String email, String password)` and `public void logOut()` methods were difficult. The code is changed various times because the code didn't run as expected in the previous iteration. So it needed some changes.

There were too many library imports and calls, so it was a bit confusing to decide which library to use according to the particular needs of the code.

5. Incompleted parts

5.1 Custom Images for Announcements

- Ability to add custom Images for announcements are not added yet.

6. References

[1] <https://www.campaignmonitor.com/css/>