Project Timeout
Software Product Manual

Revision 1.0
23.05.2015

By Team Timeout

Ali Haluk Seven	2013719111
Hakan Keçelioğlu	2012719132
Morteza Bandidarian	2013719015
Münevver Calayır	2012719021
Oğuz Çam	2013719039
Sezgi Seret	2013719108
Tuncay Altınpulluk	2013719006

Revision History

Revision	Date	Explanation
1.0	23.05.2015	Initial software product manual

Table of Contents

1. Introduction

The purpose of this software product manual is to document the implementation, testing, installation and operation of Timeout application as a software product.

Timeout Application is implemented and tested as it is described in Timeout Application Design Specification Document, Revision 2.0, satisfying the requirements in Timeout Application Requirements Specification Document, Revision 2.0, Implementation, testing and operation details are given in the following sections of this document.

2. Implementation

2.1. Source Code and Executable Organization

The source code can be downloaded from the link below.
https://code.google.com/p/swe574-timeout-project/source/browse/#git

See Section on Appendix A of this document regarding source code package and class details.

2.2. Software Tools

- Eclipse Java EE IDE for Web Developers, InteliJ IDEA and Android Studio frameworks are used for development and testing.
- Java compiler is used as compiler.
- MAMP Version 3.2.1, Apache Tomcat are used for Apache Server and MySQL Server.
- MySQL Workbench is used for database part.
- Sublime editor is used for coding .css, .html and JavaScript.

2.3. Platform

- **System software**
  Operating system: Google Cloud Compute Engine Linux VM

- **Database**
  Version: MySQL 5.5
  Region: United States
  File System Replication: Asynchronous
  Tier: D32

- **Integrated development environments**
  Name: Eclipse IDE for Java Developers
  Version: Luna Service Release 2 (4.4.2)

  Name: IntelliJ IDEA
  Version: 14.1.3

  Name: Android Studio
  Version: 1.2

3. Testing

Application should be launched in Firefox 30+, Chrome version 35+, Internet Explorer 9 and 10.

The following functions have been implemented according to the RSD and DSD and have been tested sufficiently and operating correctly.

- o Register
- o Login
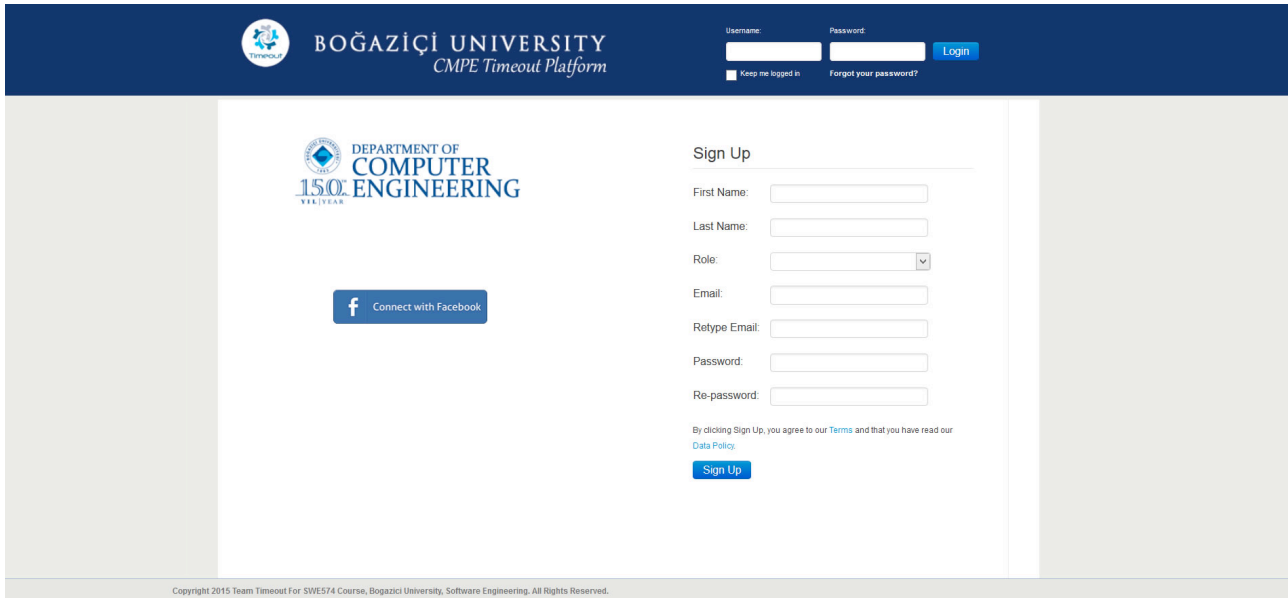- o Create Group
- o Edit Group
- o Create Event
- o Edit Event
- o Edit Profile
- o Add Comment
- o Semantic Search
- o Event Recommendation
- o Group Recommendation
- o User Recommendation
- o News Feed
- o Logout

The following functions have been implemented according to the most recent RSD and DSD but have not been tested sufficiently and not operating correctly.

- o Create Custom Type
- o Create Post
- o Edit Post
- o Edit Custom Type

The following functions have not been implemented at all.

- o Delete Event
- o Delete Comment
- o Edit Comment

4. Application Installation, Configuration and Operation

The following snapshots are user interfaces of timeout application.

Figure 2: Main Page

Figure 3: Create Event Page

Figure 4: Create Event Page with Date Picker

Figure 5: Create Event Page with Time Picker

Figure 6: Edit Profile Page

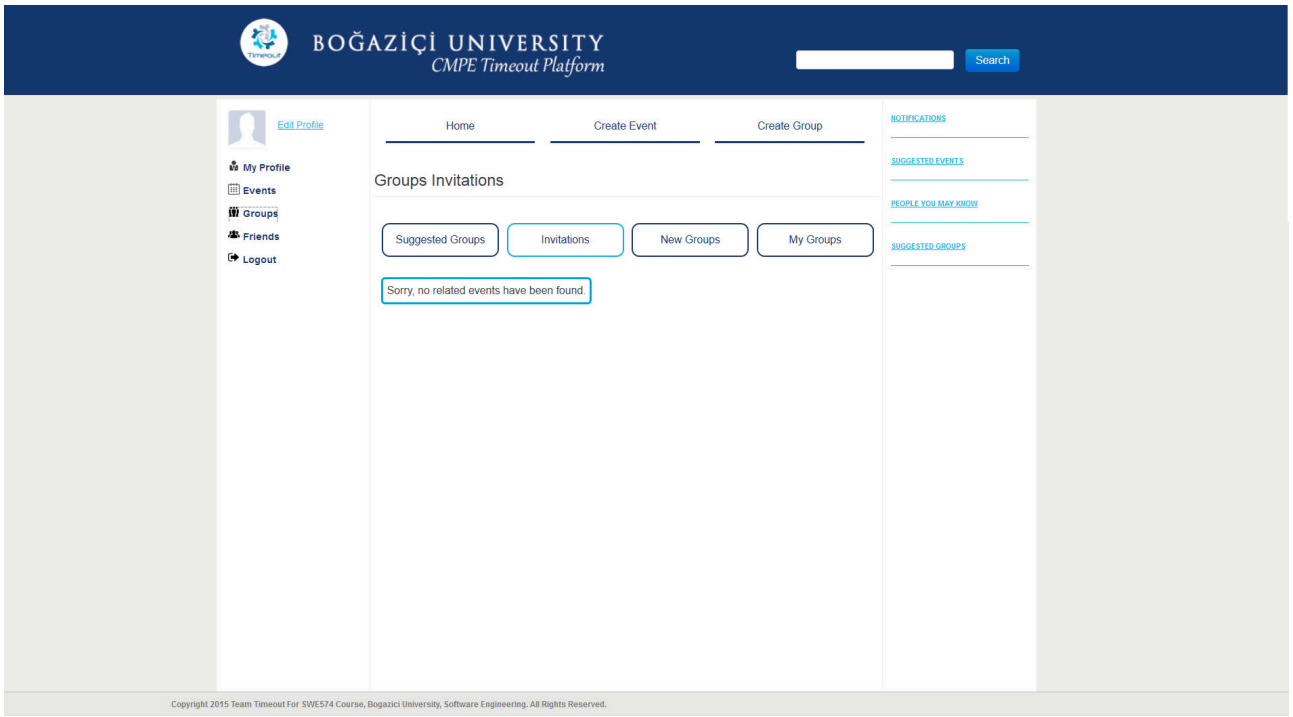

Figure 7: Edit Profile Page Photo Upload Window

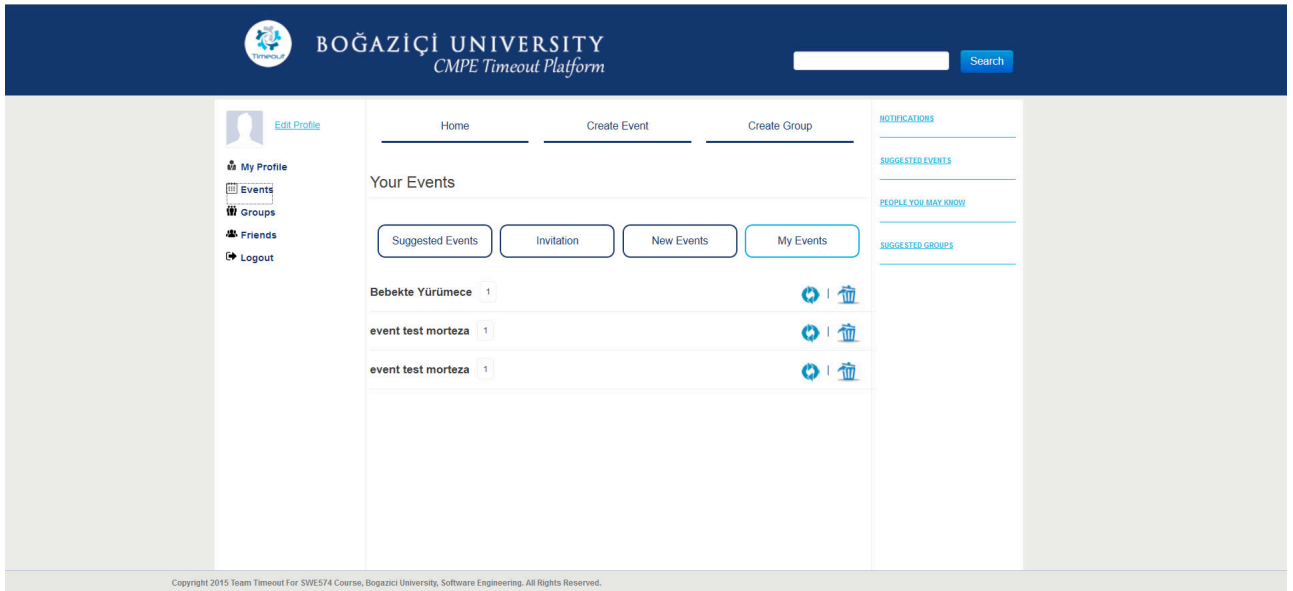**Figure 8: Group Invitation Page**

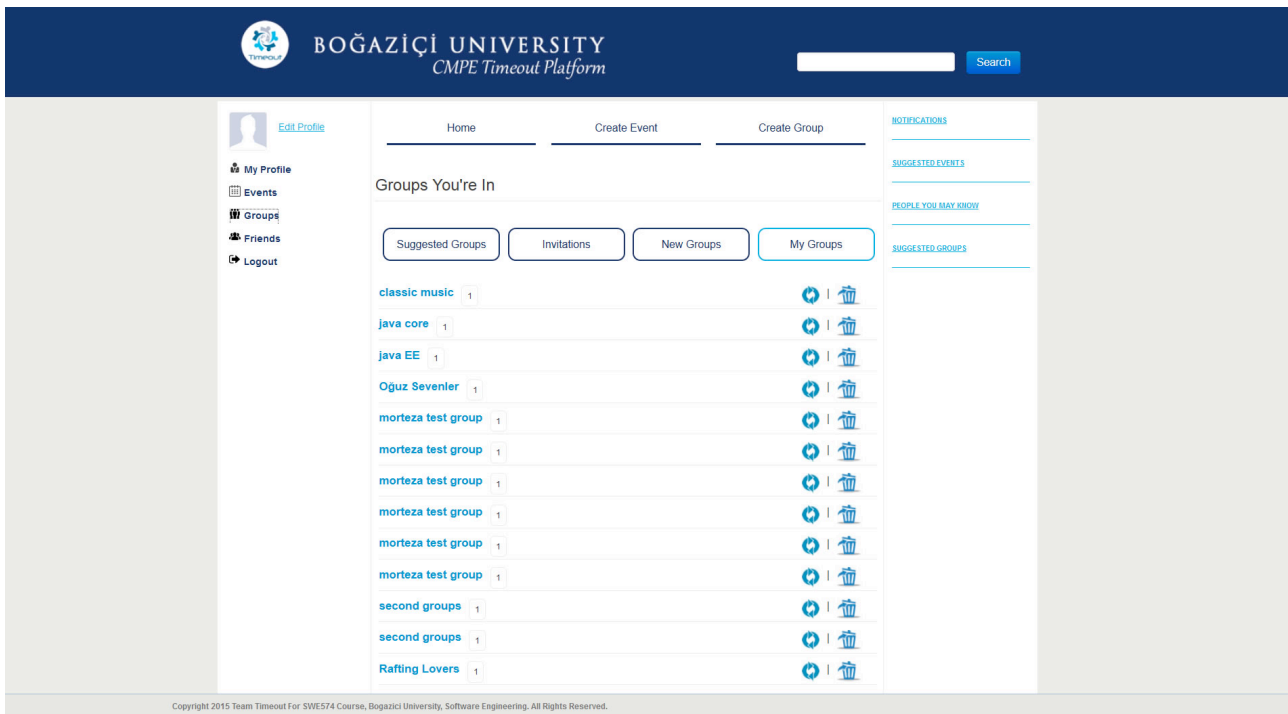**Figure 9: My Events Page**

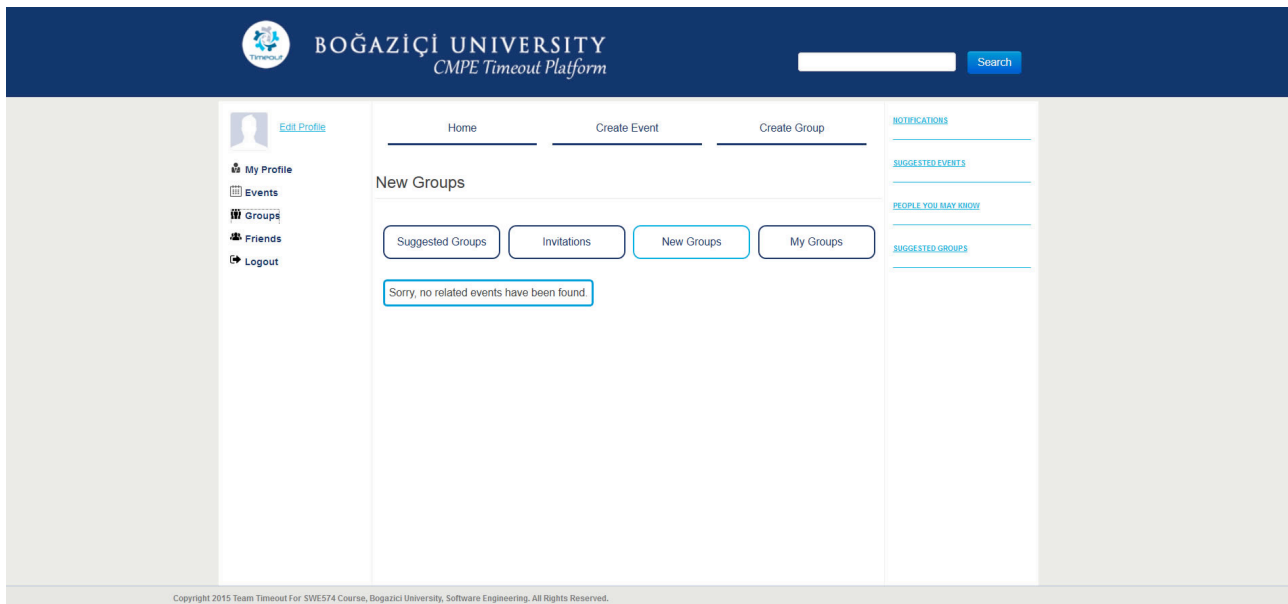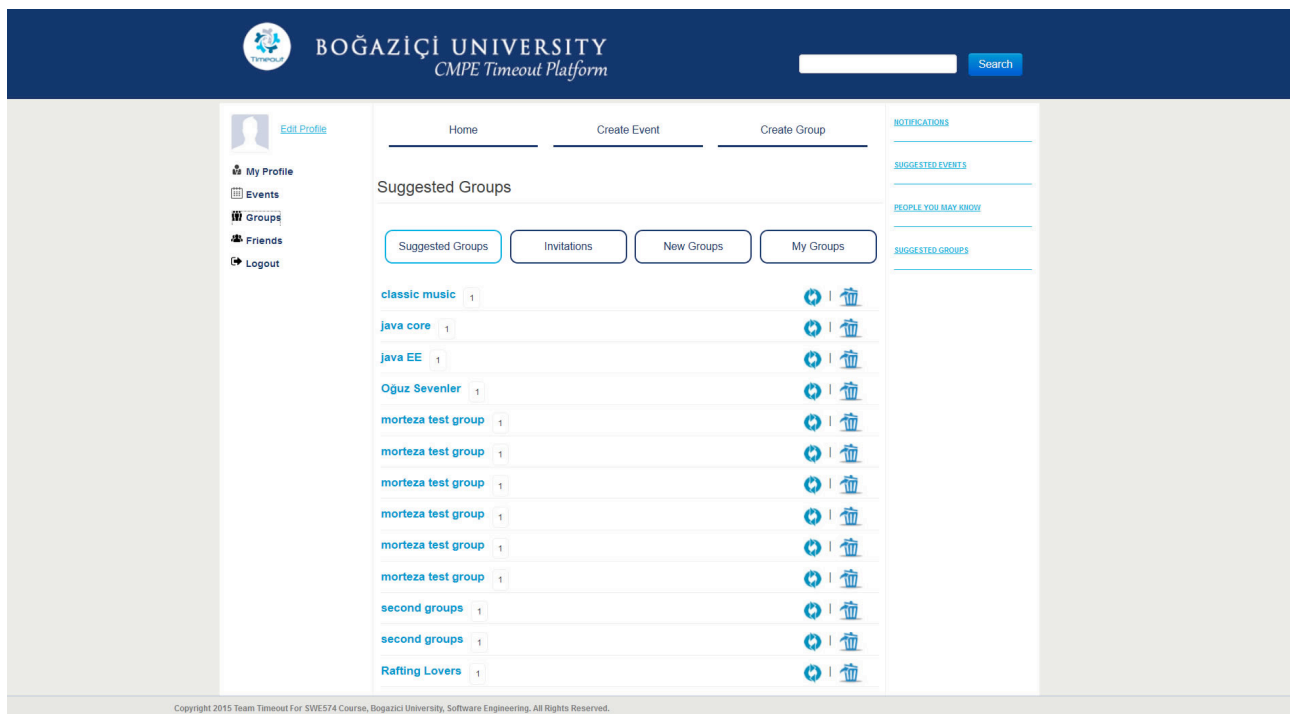**Figure 10: My Groups Page**

Figure 11: New Groups Page

Figure 12: Suggested Groups Page

Installation & Configuration:

In order to configure Google Cloud Platform,

1. Log in to your Google Account.
2. Go to your console https://console.developers.google.com/
3. Create a new Project with a Project ID and enable billing for your project.
4. Go to Cloud SQL under Storage menu.
5. Create a new instance with an Instance ID.
6. Authorize your app with assigning Project ID to Authorized App Engine applications.
7. In access control, create new user with Username: root and Password: password
8. Create a new database with following properties; Name: "demo", Character Set: utf8, Collation: utf8_general_ci

In order to configure the project in local environment and deploy it to Google Cloud,

1. Maven, JDK and MySQL 5.5 should be installed in your computer.
2. In MySQL, create a new user; Username: root and Password: password
3. Create a new database with following properties; Name: "demo", Character Set: utf8, Collation: utf8_general_ci
4. Clone project in your local directory.
5. Open appengine-web.xml file under src/main/webapp/WEB-INF directory and modify <application> with your Project ID
6. Open DBUtility under src/main/java/common and modify javax.persistence.jdbc.url with jdbc:google:mysql://your-project-id:your-instance-name/demo?user=root
7. In pom.xml properties section modify <cloudsql.url> with jdbc:google:mysql://your-project-id:your-instance-name/demo?user=root
8. Dir to /project_folder/TimeoutBackend/TimeoutBackend
9. Run 'mvn clean install' command to build project
10. Run 'mvn gae:deploy' to deploy your project to Google Cloud

11. Google Account username and password will be prompted.
12. Application is up and running in cloud.
13. Test the cloud server with sending a REST request to http://your-project-id.appspot.com/test

In order to configure the Timeout Frontend,

1. Open spa.js under TimeoutFrontEnd/public/spa.js directory and modify var backendUrl with
http://your-project-id.appspot.com/
2. Upload public folder under TimeoutFrontEnd directory to any hosting service.

References

1. GroupTimeout-ProjectTimeout-RSD-2015-05-23-Rev-2.0.docx
2. GroupTimeout-ProjectTimeout-DSD-2015-05-23-Rev-2.0.docx
3. Bennett, McRobb, Farmer, Object Oriented Systems Analysis and Design Using UML, McGraw-Hill, 2010, 4th Edition.

Appendices

Appendix A: Source Code and Executable Delivery
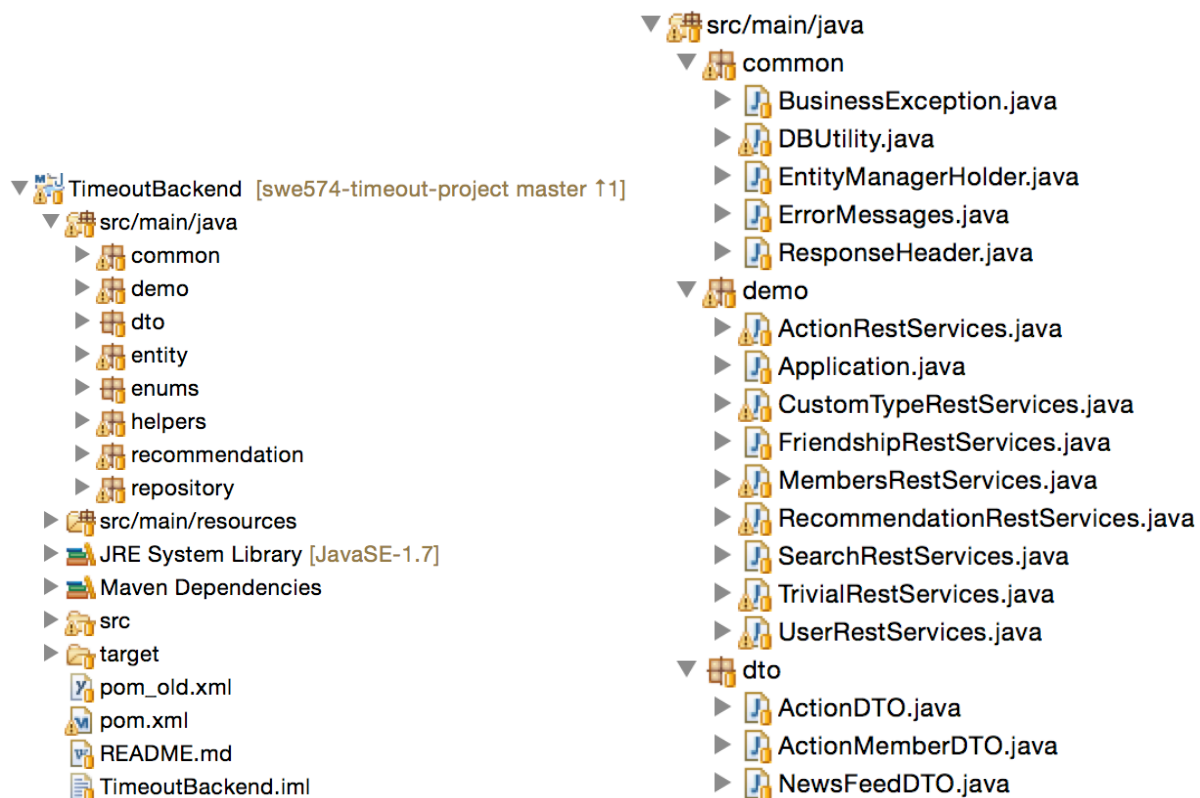
- ▼ entity
  - ▶ Action.java
  - ▶ ActionInvitation.java
  - ▶ ActionRecommendation.java
  - ▶ ActionTag.java
  - ▶ ActionUser.java
  - ▶ Attribute.java
  - ▶ AttributeValue.java
  - ▶ Comment.java
  - ▶ CustomType.java
  - ▶ Event.java
  - ▶ Friendship.java
  - ▶ Greeting.java
  - ▶ Group.java
  - ▶ Location.java
  - ▶ Notification.java
  - ▶ Post.java
  - ▶ Role.java
  - ▶ Session.java
  - ▶ Tag.java
  - ▶ User.java
  - ▶ UserBasicInfo.java
  - ▶ UserCommInfo.java
  - ▶ UserExtraInfo.java
  - ▶ UserRecommendation.java
  - ▶ UserTag.java
- ▼ enums
  - ▶ ActionType.java
  - ▶ ActionUserStatus.java
  - ▶ FeedType.java
  - ▶ FriendshipStatus.java
  - ▶ PrivacyType.java
- ▼ helpers
  - ▶ ServiceHelper.java
  - ▶ ValidationHelper.java
- ▼ recommendation
  - ▶ RecommendationEngine.java
- ▼ repository
  - ▶ ActionRepository.java
  - ▶ CustomTypeRepository.java
  - ▶ FriendshipRepository.java
  - ▶ MembersRepository.java
  - ▶ UserRepository.java