

# **System and Software Architecture Description (SSAD)**

**BlackProfessionals.net**

## **Team 6**

<b>Tian Xiang Tan</b>	Project Manager
<b>Sadeem Alsudais</b>	IIV & V
<b>Jhih-Sheng Cai</b>	System/Software Architect
<b>Aril Alok Jain</b>	Requirements Engineer
<b>Pablo Ochoa</b>	Prototyper
<b>Jeng-Tsung Tsai</b>	Operational Concept Engineer
<b>Po-Hsuan Yang</b>	Life Cycle Planner

**December 8, 2014**

# Version History

Date	Author	Version	Changes made	Rationale
10/13/14	JC	1.0	<ul style="list-style-type: none"><li>Initial Draft</li></ul>	<ul style="list-style-type: none"><li>Initial Draft for use with Instructional ICM-Sw v1.0</li></ul>
10/14/20	JC	1.1	<ul style="list-style-type: none"><li>Change of Use Case Diagram</li></ul>	<ul style="list-style-type: none"><li>Modify based on feedback from IIV &amp; V and ARB meeting</li></ul>
11/21	JC	2.0	<ul style="list-style-type: none"><li>Complete all SSAD</li></ul>	<ul style="list-style-type: none"><li>Design all system in detail.</li></ul>
11/30	JC	2.1	<ul style="list-style-type: none"><li>Change Context diagram and use cases</li></ul>	<ul style="list-style-type: none"><li>Make doc consistent base on further design and TA comment.</li></ul>
11/8	JC	2.2	<ul style="list-style-type: none"><li>Add ER-diagram</li></ul>	<ul style="list-style-type: none"><li>Add additional information</li></ul>

# Table of Contents

<b>System and Software Architecture Description (SSAD)</b> .....	<b>i</b>
<b>Version History</b> .....	<b>ii</b>
<b>Table of Contents</b> .....	<b>iii</b>
<b>Table of Tables</b> .....	<b>iv</b>
<b>Table of Figures</b> .....	<b>v</b>
<b>1. Introduction</b> .....	<b>1</b>
<b>1.1 Purpose of the SSAD</b> .....	<b>1</b>
<b>1.2 Status of the SSAD</b> .....	<b>1</b>
<b>2. System Analysis</b> .....	<b>2</b>
<b>2.1 System Analysis Overview</b> .....	<b>2</b>
<b>3. Technology-Independent Model</b> .....	<b>33</b>
<b>4. Technology-Specific System Design</b> .....	<b>34</b>
<b>4.1 Design Overview</b> .....	<b>34</b>
<b>4.2 Design Rationale</b> .....	<b>42</b>
<b>5. Architectural Styles, Patterns and Frameworks</b> .....	<b>43</b>

# Table of Tables

<i>Table 1: Actors Summary.....</i>	<i>3</i>
<i>Table 2: Artifacts and Information Summary.....</i>	<i>4</i>
<i>Table 3: Process Description.....</i>	<i>5</i>
<i>Table 4: Hardware Component Description .....</i>	<i>35</i>
<i>Table 5: Software Component Description.....</i>	<i>35</i>
<i>Table 6: Design Class Description.....</i>	<i>36</i>
<i>Table 7: Entity Relationship Description .....</i>	<i>39</i>
<i>Table 8: Architectural Styles, Patterns, and Frameworks.....</i>	<i>43</i>

# Table of Figures

<i>Figure 1: System Context Diagram .....</i>	<i>2</i>
<i>Figure 2: Artifacts and Information Diagram .....</i>	<i>4</i>
<i>Figure 3: Process Diagram .....</i>	<i>5</i>
<i>Figure 4: Hardware Component Class Diagram .....</i>	<i>34</i>
<i>Figure 5: Software Component Class Diagram .....</i>	<i>34</i>
<i>Figure 6: Deployment Diagram.....</i>	<i>35</i>
<i>Figure 7: Design Class Diagram.....</i>	<i>36</i>
<i>Figure 8: Entity Relationship Diagram .....</i>	<i>39</i>

# 1. Introduction

## 1.1 Purpose of the SSAD

The purpose of the SSAD is to document the results of the object-oriented analysis and design (OOA&D) of the system, BlackProfessionals.net, being developed.

SSAD is used for different stakeholders as followed:

1. Prentice Parr Jr., Clients, use it to help understand the structure of the system once the proposed system is delivered.
2. Prentice Parr Jr., Maintainer, use it to help understand the structure of the system once the proposed system is delivered.
3. Team 06 of CSCI577a Fall 2014, programmer (builder), as reference to the system architecture. The system being developed should be faithful to the architecture specified in the SSAD.

This document will be accompanied throughout the entire development process of the project, and will be maintained tightly. See Section 1.2 for the latest document status.

## 1.2 Status of the SSAD

The SSAD is currently at version 2.0, released by November 30, under the maintenance of Jhih-Sheng, Cai, the system/software architect of development team. The present development phase is in the Foundation phase.

## 2. System Analysis

### 2.1 System Analysis Overview

The primary purpose of the BlackProfessionals.net is to improve professional growth of people belonging African-American community. The BlackProfessionals.net can let end-users (individuals or companies) create profiles to search or be searched easily, and provide messaging function to let end-users communicate with each other directly and easily. Besides, system can generate recommendations that either people whom end-users may know or companies which they may be interested in to increase end-users' engagement opportunities. Furthermore, they can see the social feed from their friends or followed companies in the website and recent activity of them. The system can also let them invite their friends to join BlackProfessionals.net, and companies can advertise on the website.

#### 2.1.1 System Context

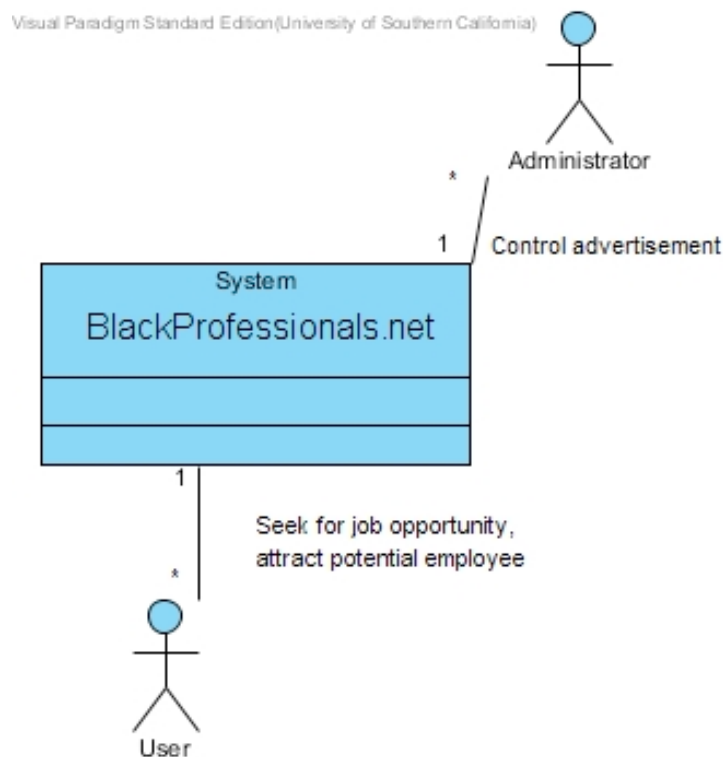


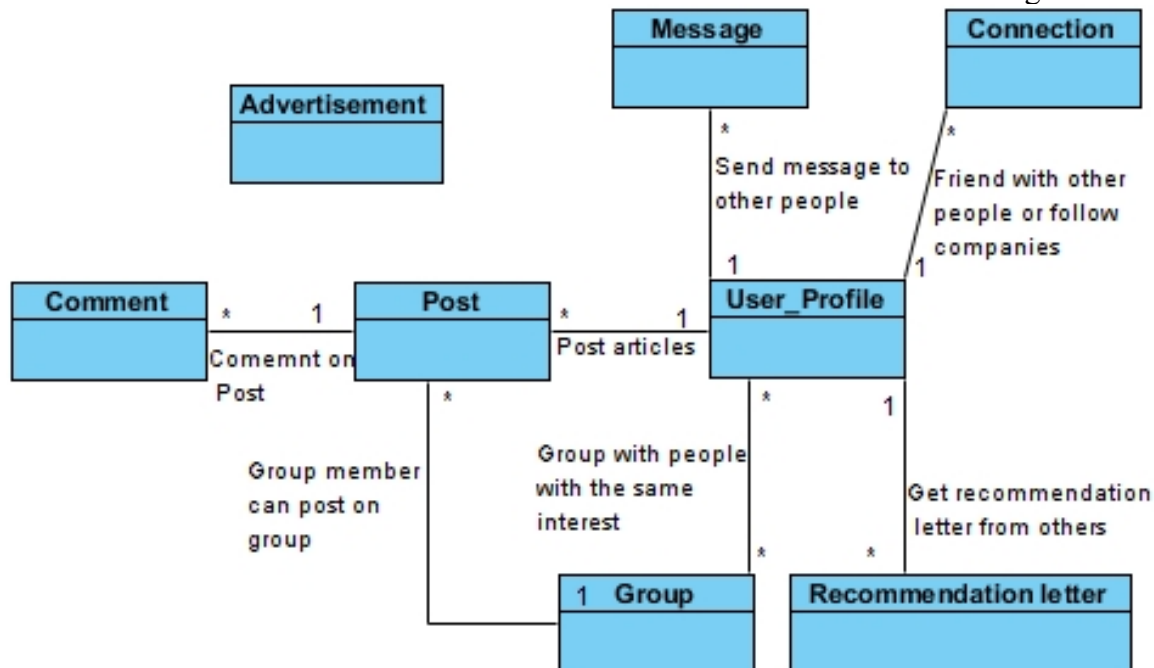
Figure 1: System Context Diagram

Actor	Description	Roles/Responsibilities
Administrator	An executive/managerial staff of this website. The administrator can approve advertisement, and modify website if needed.	<ol style="list-style-type: none"> <li>Put on/off advertisements.</li> <li>Maintain website, and modify website if needed.</li> </ol>
User (Professionals, and students)	User who come to look for guidance, jobs, or connection to others.	<ol style="list-style-type: none"> <li>Look for help or provide help.</li> <li>Find job opportunity.</li> </ol>
User (Company)	Company Representative comes to provide job information.	<ol style="list-style-type: none"> <li>Provide students and professionals with company information.</li> </ol>

Table 1: Actors Summary

## 2.1.2 Artifacts & Information

Note: The information stored in the artifact is showed in the table below the diagram.





**Figure 2: Artifacts and Information Diagram**

<b>Artifact</b>	<b>Purpose</b>
User_Profile	This table contains users information, including name, company information, skill sets, contact information, and etc.
Message	This table contains all users' pass message, including contact information and message.
Connection	This table contains users' friends list, follows companies list.
Post	This table contains all users' posts.
Comment	This table contains commends that user commend on posts.
Advertisement	This table contains the information about advertisement including company name, link, description and etc.
Recommendation Letter	This table contains recommendation history of all users, including recommender, context, date, and user information.
Group	This table contains group name and user in that group.

**Table 2: Artifacts and Information Summary**

## 2.1.3 Behavior

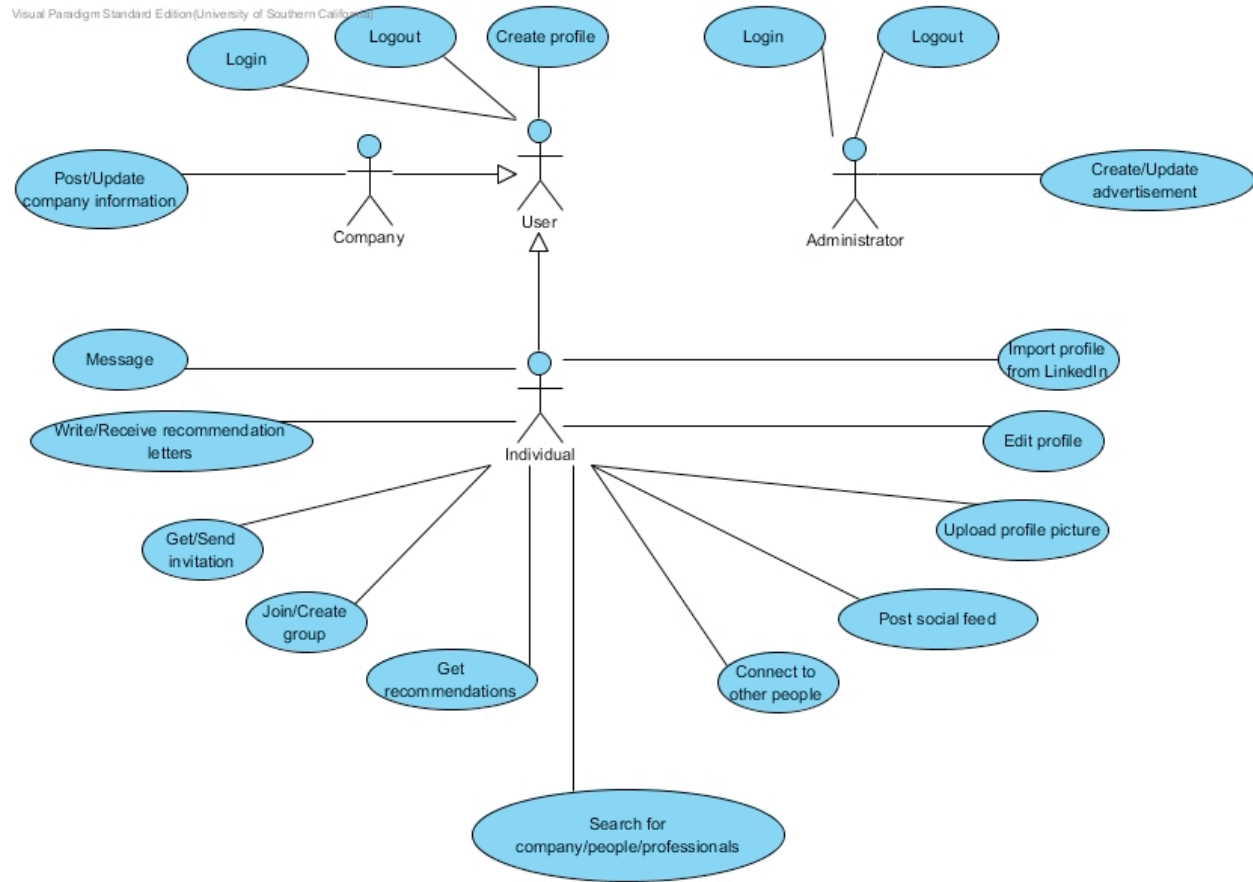


Figure 3: Process Diagram

### 2.1.3.1 Authentication

#### 2.1.3.1.1 Login

Table 3: Process Description

<b>Identifier</b>	UC-1: Login
<b>Purpose</b>	Authorize the user to log in the system and assign the user associated system role, such as students or companies.
<b>Requirements</b>	None
<b>Development Risks</b>	None
<b>Pre-conditions</b>	The user is currently on Login page.
<b>Post-conditions</b>	User is authorized, and will be given access to system operations appropriate to his/her role. Otherwise, the user is denied.

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	Enter username and password, and click login button.	
2		Send username and password to BlackProfessionals.net login checking system.
3		Redirect user to the corresponding role's main page.

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1 to 3.	
2		Redirect user to a new login page with notification of message: "Username or password is wrong. Please try again."

**2.1.3.1.2 Logout****Process Description**

<b>Identifier</b>	UC-2: Logout
<b>Purpose</b>	Log out current user and remove the sessions of current user's.
<b>Requirements</b>	None
<b>Development Risks</b>	None
<b>Pre-conditions</b>	The user has logged in and currently on any page.
<b>Post-conditions</b>	The user logouts the system.

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	Click log out button	
2		Remove the session of current user, and redirect the page to a new login page.

There is no Exceptional or Alternative Courses of action. Blank fields are treated as wrong username or password.

## 2.1.3.2 Profile management

### 2.1.3.2.1 Sign up/Join

#### Process Description

<b>Identifier</b>	UC-3: Create profile
<b>Purpose</b>	User can have present on BlackProfessionals.net
<b>Requirements</b>	WC 3393 WC 3182 WC 3181 WC 3180 WC 3169
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User click create account button in login page.
<b>Post-conditions</b>	User creates a new profile.

#### Typical Course of Action

Seq#	Actor's Action	System's Response
1	User fills in Sign-in form on landing page and clicks Join button	
2		User is shown a message requesting e-mail verification
3	User clicks on the e-mail verification link	
4		System authenticates user's e-mail address and redirects him to Build Profile page
5	User enters information in all required fields and clicks on build profile	
6		System creates a profile for the user and redirects user to "My Profile" page and displays all the information user provided

#### Alternate Course of Action

Seq#	Actor's Action	System's Response
1	Same as typical course action 1 to 4	
2	User clicks on import from "LinkedIn" button	
3		System calls LinkedIn API and a window to capture user's LinkedIn

		credentials open
4	User enters LinkedIn credentials	
5		System fetches all the required information from LinkedIn and saves it to user's profile
6		System redirects user to "My Profile" page and displays all the information received from LinkedIn

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User leaves one or more fields blank in the Sign-up form and clicks Join button	
2		System prompts user about the error and redirects user back to Landing Page with Sign-up form

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User enters new password less than 8 characters long and clicks Join button	
2		System prompts user about the error and redirects user back to Landing Page with Sign-up form

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User enters different values in new password and confirm new password during sign-up and hits Join button	
2		System prompts user about the error and redirects user back to Landing Page with Sign-up form

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User enters invalid E-mail address with invalid format	
2		System prompts user about the error

		and redirects user back to Landing Page with Sign-up form
--	--	---

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User enters e-mail address corresponding to an existing account	
2		System prompts user about the error and redirects user back to Landing Page with Sign-up form

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	Same a Alternate Course of Action 1 - 3	
2	User enters invalid/incorrect LinkedIn credentials	
3		System captures the error response from LinkedIn and displays appropriate error on the current page

**2.1.3.2.2 Create Short Summary for Profile****Process Description**

<b>Identifier</b>	UC-4: Create short summary/headline for profile.
<b>Purpose</b>	User can generate customized short summary/headline.
<b>Requirements</b>	WC_3181
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is on Edit Profile page
<b>Post-conditions</b>	User's profile is updated with new summary/headline

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	Click edit icon next to headline display	
2		System pops-up a window where user can edit the headline
3	User edits the headline as per	

	requirement and clicks on Save button at the bottom of the popped window	
4		System saves new headline and dismisses the popped-up window
5	User clicks on "Done Editing" button	
6		User is redirected to My Profile page where the newly constructed headline is shown

#### Alternative Course of Action

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1 to 2.	
2	User clicks Cancel button at the bottom of the popped up window	
3		Window is dismissed and the old headline is kept intact
4	User clicks on "Done Editing" button	
5		System redirects user to My Profile page where the user's old headline is displayed without any changes

#### Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Same a Typical Course of Action 1 - 2	
2	User removes all characters from the headline and makes it blank	
3	User hits Save button at the bottom of the window	
4		System prompts the user about the error and keeps the user on the same window

### 2.1.3.2.3 Update profile picture

#### Process Description

<b>Identifier</b>	UC-5: Upload profile picture.
<b>Purpose</b>	User can upload profile picture.
<b>Requirements</b>	WC_3169
<b>Development</b>	None

<b>Risks</b>	
<b>Pre-conditions</b>	User is on Edit Profile Page
<b>Post-conditions</b>	User's Profile is updated with new/changed profile picture

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Click edit icon on the profile picture.	
<b>2</b>		Pops-up a window where user can choose/upload profile picture
<b>3</b>	Choose required picture from current machines file system and click Save	
<b>4</b>		System saves new profile picture and dismisses the popped-up window
<b>5</b>	User clicks on "Done Editing" button	
<b>6</b>		System redirects user to My Profile page with updated profile picture

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 3	
<b>2</b>		If picture format is incorrect then display appropriate error message
<b>3</b>		Profile picture is not updated

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 3	
<b>2</b>		If picture size is larger than 4MB then display appropriate error message
<b>3</b>		Profile picture is not updated

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 2	
<b>2</b>	Users enters an invalid/incorrect filename for the picture	
<b>3</b>		System throws an appropriate error message



4		Profile picture is not updated
---	--	--------------------------------

### 2.1.3.3 Message

#### 2.1.3.3.1 Send Message

##### Process Description

<b>Identifier</b>	UC-6: Send message.
<b>Purpose</b>	User (individual only) can send message to others.
<b>Requirements</b>	WC_3170
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is in the message page and connected to others.
<b>Post-conditions</b>	Users send message to others.

##### Typical Course of Action

Seq#	Actor's Action	System's Response
1	Select "new message" button.	
2		Displays "new message" window and retrieves all the connections of the user.
3	Types message, subject, and selects receivers from the list then clicks on "send" button.	
4		Sends the message and returns to messages page.

##### Alternate Course of Action

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1 and 2	
2	Closes the "new message" window.	
3		Returns to message page.

##### Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1 and 2	
2	Doesn't select a receiver	
3		Displays a message that at least one receiver has to be selected.

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 and 2	
<b>3</b>	Leaves at least one field of "message body" or "subject" empty	
<b>4</b>		Displays a warning message and prompts the user to enter again.
<b>5</b>	User chooses either to fill the fields or leave them.	
<b>6</b>		Sends the message and returns to messages page.

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 3	
<b>4</b>		Displays a warning message and prompts the user to enter again.
<b>5</b>	User chooses either to fill the fields or leave them.	
<b>6</b>		Sends the message and returns to messages page.

**2.1.3.3.2 View sent Messages****Process Description**

<b>Identifier</b>	UC-7: View sent messages.
<b>Purpose</b>	User views sent messages to others.
<b>Requirements</b>	WC_3179
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is in the message page.
<b>Post-conditions</b>	User views sent messages to others.

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Select "sent" button.	
<b>2</b>		Retrieves sent messages and displays

		sent messages page.
--	--	---------------------

**Alternate Course of Action**

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1	
2		User doesn't have sent emails. They system displays a message, and displays messages page.

**2.1.3.4 Recommendation****2.1.3.4.1 Get recommendation****Process Description**

<b>Identifier</b>	UC-8: Get recommendation
<b>Purpose</b>	Generate recommendations of users(individual only) to connect with
<b>Requirements</b>	WC 3173
<b>Development Risks</b>	Risk of creating an accurate recommendation algorithm is high
<b>Pre-conditions</b>	User is logged in
<b>Post-conditions</b>	None

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User clicks the home page link	
2		System shuffles recommendations in random order
3		System displays recommendations on screen

**Alternate Course of Action**

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1	
2		System runs similarity algorithm and finds nothing
3		System displays no recommendations found on the display

## 2.1.3.5 Search and Connection

### 2.1.3.5.1 General Search Functionality

#### Process Description

<b>Identifier</b>	UC-9: General Search functionality
<b>Purpose</b>	User can search other users based on certain parameters
<b>Requirements</b>	WC_3174, WC_3183, WC_3420, WC_3421, WC_3423, WC_3424, WC_3445
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is logged-in
<b>Post-conditions</b>	User gets a list of results based on search parameters

#### Typical Course of Action

Seq#	Actor's Action	System's Response
1	User types in a "string" (which is expected to be the name of the people, company or group) in the search bar at the top of any page	
2		Check whether the input is empty or not. If the input is not empty, enable the search button.
3	User click the search button	
4		Validate the input, do the SQL injection protection on the input and then use the string to get the matching data from the database, and return the list of result to the user.

#### Alternate Course of Action

Seq#	Actor's Action	System's Response
1	User types in a "string" (can be job title, another users name, company/school name, other attributes) in the search bar at the top of any page and hits Search button	
2		If there are no matching results or the result is a private group, system shows a message "No Results Found" on the result page

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User tries to hit the search button after putting an empty string in search bar	
2		Search button is disabled till a non-empty string is entered in search bar

**2.1.3.5.2 Advanced Search Functionality****Process Description**

<b>Identifier</b>	UC-10: Advanced Search functionality
<b>Purpose</b>	User can search other users based on certain parameters
<b>Requirements</b>	WC_3174, WC_3183, WC_3420, WC_3421, WC_3423, WC_3424, WC_3445
<b>Development Risks</b>	Implementing all the advanced search options the clients expects and making them part of search algorithm
<b>Pre-conditions</b>	User is logged-in
<b>Post-conditions</b>	User gets a list of results based on search parameters

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User presses the advanced search button	
2		User is redirected to an advanced search page for people where several options to fine-tune the search are available and an empty list of search result.
3	User select the type he/she wants to search(to search people, company or group).	
4		User is redirected to an advanced search page for the type he/she selected where several options to fine-tune the search are available and an empty list of search result.
5	User fills in the given fields in the search fields.	
6		Check whether there is a non-empty filed in the search fields. If there is one,

		then enable the search button.
7	User click the search button	
8		Validate the input, do the SQL injection protection on the input and then use the string to get the matching data from the database, and return the list of result to the user.

**Alternate Course of Action**

Seq#	Actor's Action	System's Response
	Same as steps 1-3 of Typical course of action	
4		If there are no matching results or the result is a private group, system shows a message "No Results Found" on the result page

**Exceptional Course of Action**

Seq#	Actor's Action	System's Response
	Same as steps 1-2 of Typical course of action	
3	User enters an empty-string of keywords	
4		Search button is disabled till a non-empty string is entered in search fields.

**2.1.3.5.3 User Connection****Process Description**

<b>Identifier</b>	UC-11: User (individual only) send connection request
<b>Purpose</b>	User (individual only) can connect to other users.
<b>Requirements</b>	WC_3168
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User log in the system
<b>Post-conditions</b>	User send a connection request to them

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User press 'connect' button on other user introduction page or	

	in the recommendation area.	
2		System records the request and refresh the page which contain a disabled button with rename value: Request Sent

#### 2.1.3.5.4 User Accept/Reject Connection

##### Process Description

<b>Identifier</b>	UC-12: User (individual only) accept/reject connection
<b>Purpose</b>	User (individual only) can connect to other users.
<b>Requirements</b>	WC_3492
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User log in the system
<b>Post-conditions</b>	User connect/not connect to the other user

##### Typical Course of Action

Seq#	Actor's Action	System's Response
1	User press 'accept' button in the connection list	
2		System records the connection between two users and change the status on the page to the value 'approved'.

##### Alternate Course of Action

Seq#	Actor's Action	System's Response
1	User press 'reject' button in the connection list	
2		System will remove the request from the connection list.

#### 2.1.3.5.5 User Cancel Connection

##### Process Description

<b>Identifier</b>	UC-13: User (individual only) cancel connection
<b>Purpose</b>	User (individual only) can connect to other users.
<b>Requirements</b>	WC_3493

<b>Development Risks</b>	None
<b>Pre-conditions</b>	User log in the system
<b>Post-conditions</b>	User disconnect with the other user

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User press 'remove' button in his/her connection list.	
2		A dialog show up with message 'Are you sure to remove the connection?'
3	User click 'yes' button	
4		System will remove the connection between the users and the user will be removed from the connection list.

**Alternate Course of Action**

Seq#	Actor's Action	System's Response
	Same as steps 1-2 of Typical course of action	
3	User click 'no' button	
4		The dialog will disappear.

**2.1.3.5.6 User follow/Unfollow a company****Process Description**

<b>Identifier</b>	UC-14: Follow/Unfollow a company
<b>Purpose</b>	User (individual only) can follow a company in the website.
<b>Requirements</b>	WC_3491
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User has logging the system
<b>Post-conditions</b>	User follow/unfollow the company and can/cannot receive the social feed, message from the company.

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User press the 'follow' button in the company introduction page.	



2		The button will be turned into 'follow' and the system will record the user has followed the company.
---	--	---

#### Alternative Course of Action

Seq#	Actor's Action	System's Response
1	User press the 'unfollow' button in the company introduction page.	
2		The button will be turned into 'follow' and the system will record the user has unfollowed the company.

## 2.1.3.6 Advertisement

### 2.1.3.6.1 Manage Advertisement

#### Process Description

<b>Identifier</b>	UC-15: Manage Advertisement.
<b>Purpose</b>	Administrator can create advertisement for application.
<b>Requirements</b>	WC_3176 WC_3448
<b>Development Risks</b>	None
<b>Pre-conditions</b>	The logged in user is an admin
<b>Post-conditions</b>	User create an advertisements to be showed in website.

#### Typical Course of Action

Seq#	Actor's Action	System's Response
1	Admin clicks on the manage Ads link	
2		System shows existing ads page
3	Admin clicks the New ad button	
4		System shows the Ad detail page
5	Admin enters the ad description, content and clicks the Save button	
6		System show the existing ads page with the new ad

**Alternate Course of Action 1**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 2	
<b>3</b>	Admin clicks the Edit button for one of the ads	
<b>4</b>		Systems show the Ad detail page filled with the existing ad details
<b>5</b>	Admin edit the ad content and clicks the Save Button	
<b>6</b>		System show the existing ads page with the changed ad

**Alternate Course of Action 2**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 2	
<b>3</b>	Admin clicks the Delete button	
<b>4</b>		System shows a popup asking for confirmation before deleting the ad
<b>5</b>	Admin clicks the Yes button	
<b>6</b>		System show the existing ads page without including the deleted ad

**Alternate Course of Action 3**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as step 1 to 4 of alternative course of action 2	
<b>3</b>	Admin clicks the Cancel button	
<b>4</b>		System shows the existing ads page without doing any change to the ads

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as steps 1-4 of typical course of action	
<b>2</b>	Admin clicks the Save button	
<b>3</b>		System shows error message in red, saying that "You didn't provide complete information"

### 2.1.3.7 Recommendation letter

#### 2.1.3.7.1 Write recommendation letter

##### Process Description

<b>Identifier</b>	UC-16: Write recommendation letter
<b>Purpose</b>	User (individual only) can write recommendation letter to others
<b>Requirements</b>	WC 3184
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is in others' profile page
<b>Post-conditions</b>	Users submit a recommendation letter to others

##### Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click "write recommendation letter to your friend" button	
2		System redirect to write recommendation letter page.
3	User fulfill the required context and click submit button	
4		System records the record, and send message to this user.

##### Alternate Course of Action

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1 to 2	
3	User fulfill the required context and click cancel button	
4		Return to other's profile page.

##### Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Same as typical course of action 1 to 3	
2		System can't find user and return user does not exist.

### 2.1.3.7.2 Accept/Reject recommendation letter

#### Process Description

<b>Identifier</b>	UC-17: Accept/Reject recommendation letter.
<b>Purpose</b>	User (individual only) can accept or reject recommendation letter from others.
<b>Requirements</b>	WC 3184
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is in his/her own recommendation letter page.
<b>Post-conditions</b>	Users show the recommendation letters they want to show others.

#### Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click which recommendation letter they want to show to others, and click apply button	
2		System records the record and displays on his/her profile page.

#### Alternative Course of Action

Seq#	Actor's Action	System's Response
1	Click which recommendation letter they want to show to others, and click cancel button	
2		System returns to recommendation letter page.

### 2.1.3.8 Group

#### 2.1.3.8.1 Create group

#### Process Description

<b>Identifier</b>	UC-18: Create group
<b>Purpose</b>	User (individual only) can create groups
<b>Requirements</b>	WC_3442
<b>Development Risks</b>	None

<b>Pre-conditions</b>	User is on create group page
<b>Post-conditions</b>	User is on new group page

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User fill create group form, and click create button	
<b>2</b>		Create group in database and redirect to group page.

**Alternate Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User file create group form, but click cancel button	
<b>2</b>		System redirect user to group list page

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User fill create group form, and click create button	
<b>2</b>		System detect that some column violate restriction, and show dialog to request user to refill some column value.
<b>3</b>	User refill requested column and click submit button.	
<b>4</b>		User fill create group form, and click create button

**2.1.3.8.2 Group management****Process Description**

<b>Identifier</b>	UC-19: Group management
<b>Purpose</b>	Individual user with admin privilege for the group can manage group status
<b>Requirements</b>	WC_3483
<b>Development Risks</b>	None
<b>Pre-conditions</b>	Group admin visit group about page.
<b>Post-conditions</b>	Group policy and member list is updated

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Group admin approve/decline applicants to joint the group	
<b>2</b>		System record the changes and refresh about page

**Alternative Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User click "Edit" button on about page	
<b>2</b>		System redirect to edit group setting page
<b>3</b>	Group admin change the status of group setting and click submit button.	
<b>4</b>		System record the changes and redirect to group "about" page

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Group admin change the status of group setting or approve/decline the users' application to join group	
<b>2</b>		System detects that group name column is empty or invalid, and display red word on top of page: "please fill in required column value"
<b>3</b>	Group admin fill in required column and press submit button	
<b>4</b>	Same as Typical course of action 2 to 4	

**2.1.3.8.3 Join/Leave group****Process Description**

<b>Identifier</b>	UC-20: Join/Leave group
<b>Purpose</b>	User (individual only) can join or leave group anytime
<b>Requirements</b>	WC 3444, WC 3446
<b>Development Risks</b>	None

<b>Pre-conditions</b>	User is on group page
<b>Post-conditions</b>	Group page will show correspond page based on users request.

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User press join group/ leave group button	
2		System pops up dialog that asks user if they want to join or leave group.
3.	User press yes button	
4.		System record action and show correspond status changes in group page("Request for joining group" or "Join group")

**Alternative Course of Action**

Seq#	Actor's Action	System's Response
1	Same as steps 1-2 of Typical course of action	
3	User clicks no button.	
4		System stay in current page.

**2.1.3.9 Invitation****2.1.3.9.1 Invite friends through email****Process Description**

<b>Identifier</b>	UC-21: Invite friends through email
<b>Purpose</b>	User (individual only) can invite friends through email
<b>Requirements</b>	WC_3175
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User should login
<b>Post-conditions</b>	None

**Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User input their friend's email	

	address and click “invite”	
2		Check whether user’s input is an valid email address or not
3		Check whether this email has already in our system or not
4		Send an invitation email
5		Check whether the email is sent or not
6	Show invitation page	

**Exceptional Course of Action**

Seq#	Actor’s Action	System’s Response
1	Same as typical course of action 1	
2		System find that input email is not a valid email, and return to invitation page

**Exceptional Course of Action**

Seq#	Actor’s Action	System’s Response
1	Same as typical course of action 1 to 2	
3		System find that input email is already in our system, and return to invitation page

**Exceptional Course of Action**

Seq#	Actor’s Action	System’s Response
1	Same as typical course of action 1 to 5	
6		System find out email is not sent, and return to invitation page

**2.1.3.9.2 Invite friends from LinkedIn****Process Description**

<b>Identifier</b>	UC-22: Invite friends from LinkedIn
<b>Purpose</b>	User (individual only) can invite friends from LinkedIn
<b>Requirements</b>	WC_3175
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User should login



<b>Post-conditions</b>	None
------------------------	------

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User click "invite" button on menu bar	
<b>2</b>		Go to "get friend list " controller
<b>3</b>		Invoke LinkedIn API
<b>4</b>		LinkedIn will ask user to input email and password
<b>5</b>	User input email and password	
<b>6</b>		LinkedIn return friend list
<b>7</b>		Check the who is not in our system yet
<b>8</b>		Block the original page and then show a dialog with many checkbox
<b>9</b>	Select the checkboxes that user want to invite	
<b>10</b>		Send an invitation email
<b>11</b>		Check whether the email is sent or not
<b>12</b>		Unblock the page
<b>13</b>	Show invitation page	

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 5	
<b>6</b>		System detect that user input email and password to LinkedIn are wrong , and return to invitation page

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course of action 1 to 11	
<b>12</b>		System detect that email is not sent, and return to invitation page

**2.1.3.9.3 Invite friends into group****Process Description**

<b>Identifier</b>	UC-23: Invite friends to group
-------------------	--------------------------------

<b>Purpose</b>	User (individual only) can invite friends in our platform to one group
<b>Requirements</b>	WC_3443
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User should login and also should be in group
<b>Post-conditions</b>	None

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
1	User click "invite" button on menu bar	
2		Check your connection/friend list to see who is not in the group
3		Block the original page and then show a dialog with many checkbox
4	Select the checkboxes that user want to invite	
5		Send an invitation/notification in our platform
6		Check whether an invitation is sent or not
7		Unblock the page
8	Show the group home page	

**Exceptional Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
1	Same as typical course of action 1 to 6	
7		System detect that invitation is not sent, and return to group home page

**2.1.3.10 Social Feed****2.1.3.10.1 Post an update****Process Description**

<b>Identifier</b>	UC-24: Post an update
<b>Purpose</b>	User can post an update
<b>Requirements</b>	WC_3172
<b>Development</b>	None

<b>Risks</b>	
<b>Pre-conditions</b>	User is on home page
<b>Post-conditions</b>	The update record is created and will show up as the first social feed on the page.

#### Typical Course of Action

Seq#	Actor's Action	System's Response
1	User clicks on tab to choose the type of update	
2		System shows the corresponding fields for the specific type
3	User types in the content and press "Post"	
4		System validates the fields
5		System creates a new update record with the entered information
6		System shows the social feed with the latest update in chronological order

#### Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Same as steps 1-2 of Typical course of action	
3	User doesn't type in all the fields	
4		System shows error message in red, saying that "You didn't provide complete information."

### 2.1.3.10.2 View social feed

#### Process Description

<b>Identifier</b>	UC-25: View social feed
<b>Purpose</b>	User can view his connection latest activities
<b>Requirements</b>	WC_3489
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User is logged into the system
<b>Post-conditions</b>	None

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User clicks the Home link on top bar of the webpage	
<b>2</b>		Systems retrieves user connections activities
<b>3</b>		System shows separate fields based on type of the activity

**Alternate Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as typical course action 1 to 2.	
<b>3</b>		System retrieves user connections activities and finds nothing
<b>4</b>		System shows a "No recent activity" message on home page

**2.1.3.10.3 Comment on a post****Process Description**

<b>Identifier</b>	UC-26: Comment on a post
<b>Purpose</b>	User can post a social feed
<b>Requirements</b>	WC_3422
<b>Development Risks</b>	None
<b>Pre-conditions</b>	User views updates in the social feed
<b>Post-conditions</b>	The comment record is created and appears at the end of the comment list of an update

**Typical Course of Action**

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	User clicks on "Comment" link corresponding to the update to be commented on	
<b>2</b>		System displays the comments that are originally existing, in chronological order

<b>3</b>	User types in the message and clicks on "Comment" button.	
<b>4</b>		System records the comment
<b>5</b>		System displays the latest comments in chronological order

#### Alternate Course of Action

<b>Seq#</b>	<b>Actor's Action</b>	<b>System's Response</b>
<b>1</b>	Same as step 1 of Typical course of action	
<b>2</b>		System displays "There is no comment made on this update" underneath the corresponding update

### 2.1.4 Modes of Operation

BlackProfessionals.net will operate in only one mode, so there is no additional information concerning modes of operations provided.

### **3. Technology-Independent Model**

BlackProfessionals.net does not have technology-independent model since our system is designed on technologies available today. That is, many technologies such as developing languages, web server, database manager, have already chosen by development team and agree by client. Therefore this section is intentionally omitted.

## 4. Technology-Specific System Design

### 4.1 Design Overview

#### 4.1.1 System Structure

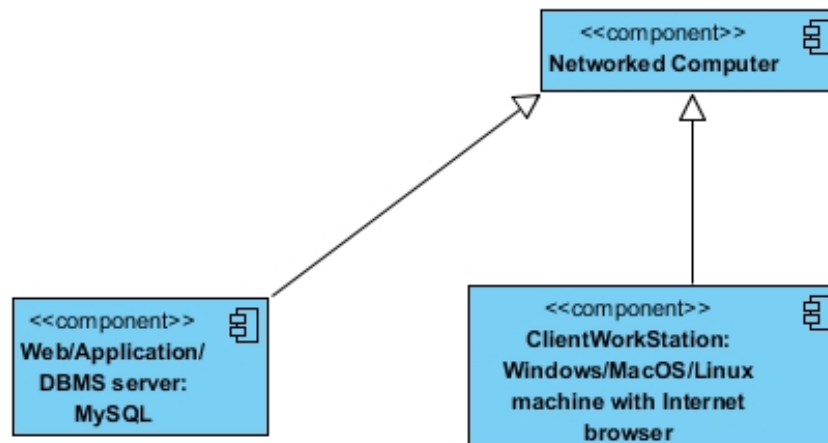


Figure 4: Hardware Component Class Diagram

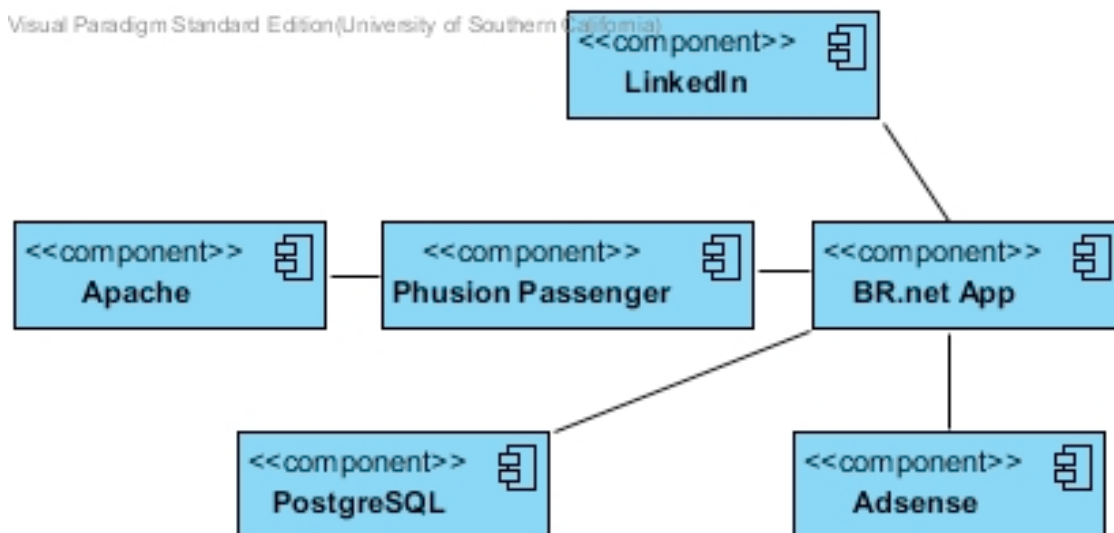


Figure 5: Software Component Class Diagram

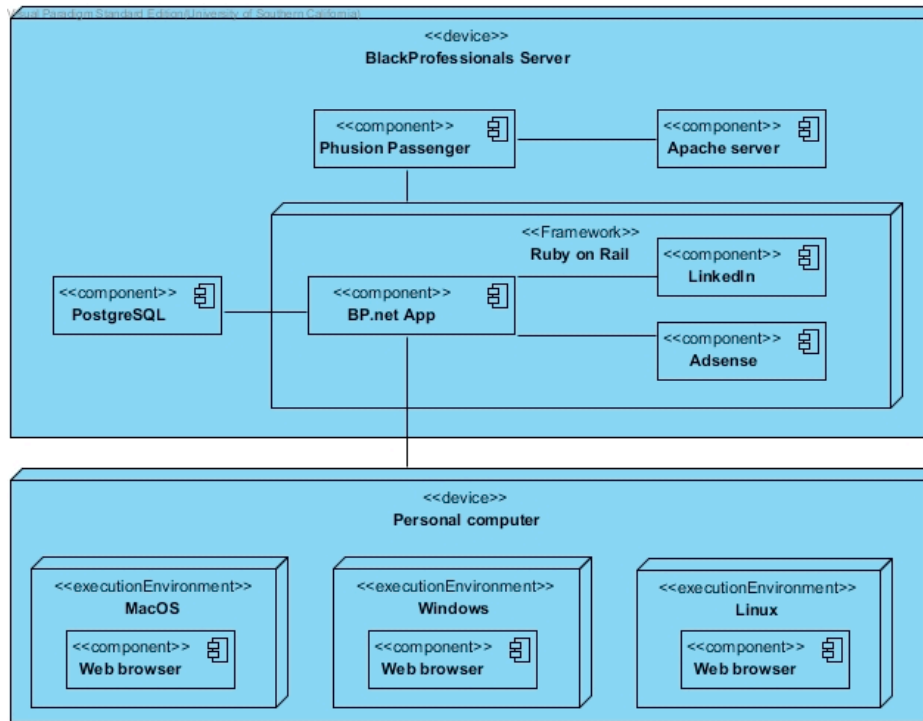


Figure 6: Deployment Diagram

Table 4: Hardware Component Description

Hardware Component	Description
Web/application/DBMS Server	This component accepts connection from all users.
Networked computer	A computer that is connected to other networked computer through internet.
Client Workstation	This is a user computer. It must have a web browser and must have Internet connection.

Table 5: Software Component Description

Software Component	Description
PostgreSQL	The DBMS for the website.
Phusion Passenger	The application server for the project.
Apache	The http server for the project, use Rail as framework.
AdSense	The advertisement program for the project.
LinkedIn(ruby)	To get user's profile from LinkedIn
BP.net App	To run BlackProfessionals.net application in Ruby on rails.



## 4.1.2 Design Classes

### 4.1.2.1 Class diagram

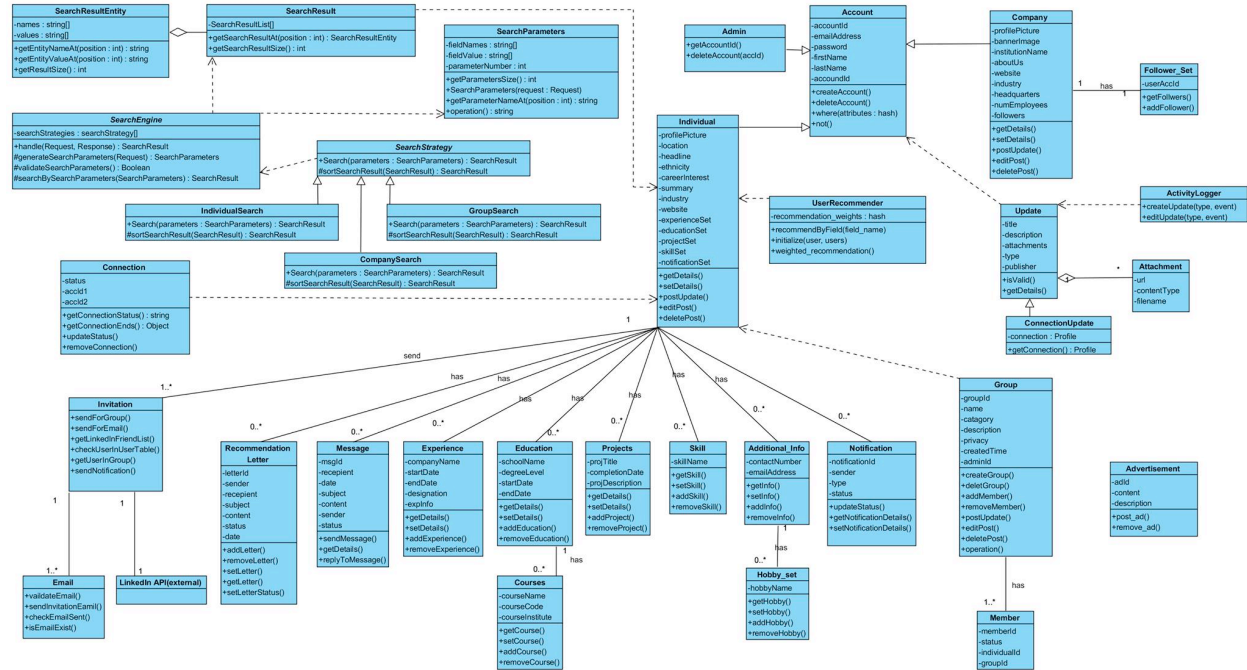


Figure 7: Design Class Diagram

Table 6: Design Class Description

Class	Type	Description
Account	Entity	All users in our system include admin, individuals, and companies.
Admin	Entity	Administrator of this system
Individual	Entity	Individual include students and professionals.
Experience	Boundary	This component records work experience for users in the past.
Education	Boundary	This component has all education records of users, and enables the access to them.
Courses	Boundary	This component has all course records of users, and enables the access to them.
Projects	Boundary	This component has all project records of users, and enables the access to them.
Skill_Set	Boundary	This component has all skill records of users, and enables the access to them.
Addtional_Info	Boundary	This component has all additional

		information user have, and enables the access to them.
Notification	Boundary	This component has all notification from connection if users in connection update activities.
Message	Boundary	This component has all the messages that sent to or receive from others, and enable user to send message.
Invitation	Boundary	This component enable user to send invitation to others to join their network (friends, group, or BPnet) for internal or external users.
Email	Boundary	This component enable user to check email address not only the status, and correctness, but if it exists in our system.
LinkedIn	Boundary	This component enable user to send invitation to user's network in LinkedIn.
Connection	Boundary	This component enable user to record their connection status with others, and enable them to change it.
Hobby_set	Boundary	This component has all hobbies of users, and enables the access to them.
Recommendation letter	Boundary	This component has all the recommendation letters that are sent to or received from others, and enable user to manages the letters.
Group	Entity	This component records all group info in the system, and enables the access to them.
MemberList	Controller	This component records the member list for a group, and enables the access to them.
UserRecommender	Boundary	This component enable fetching the recommended connections based the weighted score.
Company	Entity	This component includes companies and headhunters.
Follower_Set	Controller	This component record company followers' info.
Update	Controller	This component shows the social feed for user.
ActivityLogger	Controller	This component allows users to create/edit an activity.
Attachment	Entity	This component record social feed attachment info.
ConnectionUpdate	Controller	This component get social feed publisher's info and publisher's connection info.
SearchResultEntity	Entity	This component refers to a certain entry of

		search results, and enables getting data in terms of Name or Value.
SearchResult	Boundary	This component refers to the whole search results, which are consist of a number of SearchResultEntity.
SearchStrategy	Boundary	This component refers to the whole search strategies, which are the generalization of different types of searches.
GroupSearch	Boundary	This component refers to the search strategy specific for searching groups.
IndividualSearch	Boundary	This component refers to the search strategy specific for searching individuals.
CompanySearch	Boundary	This component refers to the search strategy specific for searching companies.
SearchParameters	Boundary	This component refers to the parameters utilized by search engine.
SearchEngine	Controller	This component refers a search engine that can prepare parameters, searches accordingly and produces the results.
Advertisement	Entity	This component has all advertisements admin adds to show to potential customers.

### 4.1.2.2 Entity- Relationship Diagram

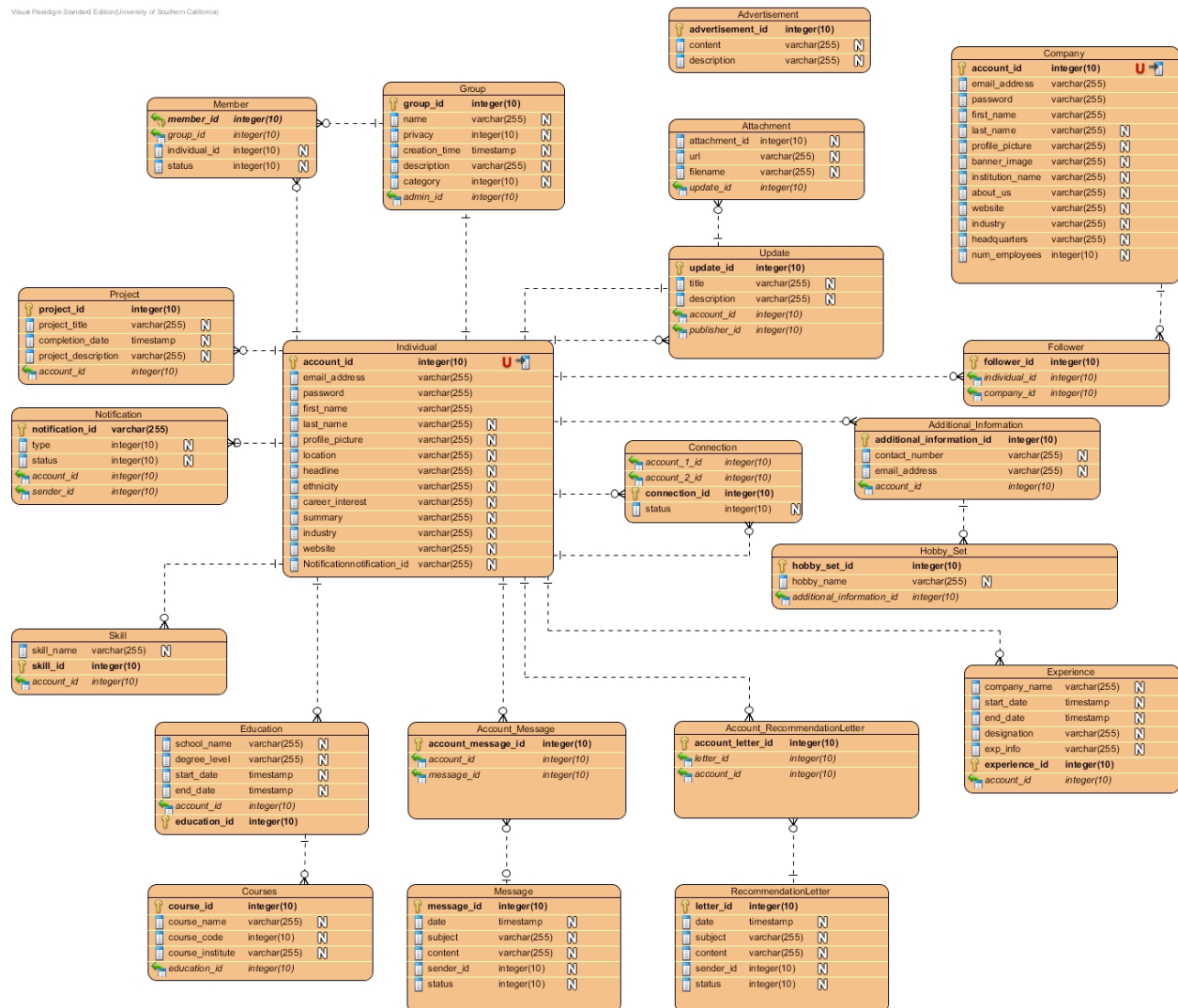


Figure 8: Entity Relationship Diagram

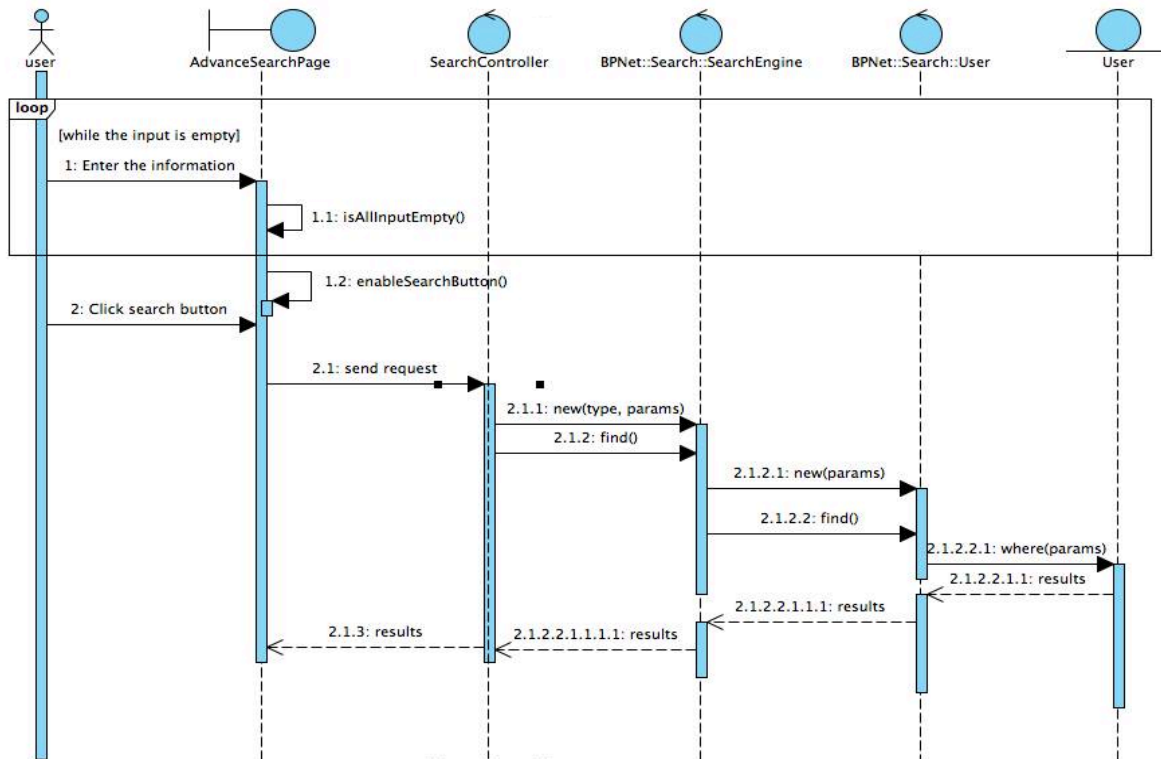
Table 7: Entity Relationship Description

Class	Description
Individual	This table records all the information for individual. Individual include students and professionals.
Group	This table records group information.
Member	This table records the relationship between Individual and Group.
Project	This table contains the project individual participated.
Experience	This table records work experience for users in the past.
Education	This table records the education history of user.

Course	This table records all course records of users.
Skill	This table records all skills users have.
Connection	This table record users connection status with others, and enable them to change it.
Notification	This table records all notification from connection if users in connection update activities.
Message	This table records all the messages that sent to or receive from others, and enable user to send message.
Account_Message	This table records the relationship between message and individuals.
Additional information	This table records all user additional information.
Hobby_Seet	This table records all hobbies of users, and enables the access to them.
Company	This table records companies and headhunters information.
Follower	This table records the relationship between company and individual.
Attachment	This table records social feed attachment info.
Update	This table records social feed publisher's info and publisher's connection info.
RecommendationLetter	This table records all the recommendation letters that are sent to or received from others, and enable user to manages the letters.
Account_Recommendation	This table records the relationship between account and individual.
Advertisement	This table records all advertisements admin adds to show to potential customers.

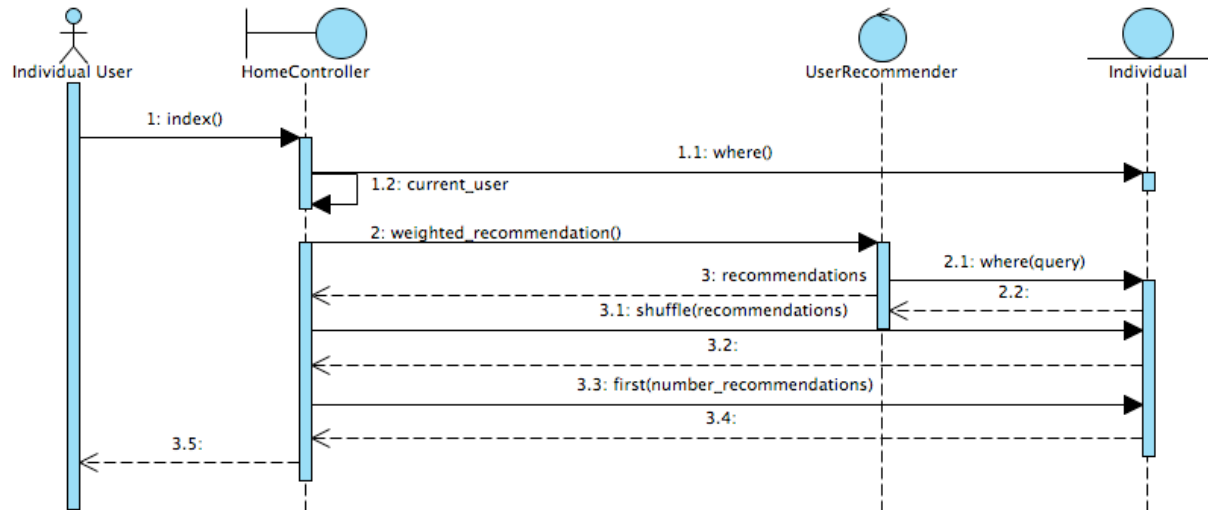
### 4.1.3 Process Realization

#### 4.1.3.1 Search Engine



**Figure 9: Search engine sequential Diagram**

### 4.1.3.2 Recommendation Engine

**Figure 9: Recommendation engine sequential Diagram**

## 4.2 Design Rationale

In BlackProfessionals.net, we aim to create another LinkedIn for black professionals. In order to make all users (Professionals, Students, and Companies) connect with each other without any regional boundary, it must be accessible via the Internet for all users. Therefore, BlankProfessionals.net is designed as a web application.

In database design, we separate users (individual, group, admin) into three groups for different individual behaviors. For individual, it contains professionals, students, and since this is for most of our users, all design is surrounded by it, even the connection between two different kinds of users is designed in individual.

Since this web application is internet-based, in order to make system more flexible and easy to change in the future, we chose Model-View-Control (MVC) as our architecture pattern. This pattern allows us to separate three parts of the system:

1. Model (database, which stores the value of system)
2. View (User interface)
3. Control (Server, which will control the status of model and view)

Since the connection between MVC model can be implemented by client and server style easily, and client and server style can extend the system easily, we chose client and server style as architecture style.

In our Model, we use PostgreSQL as our server since it is the most advanced open source SQL server.

## 5. Architectural Styles, Patterns and Frameworks

**Table 8: Architectural Styles, Patterns, and Frameworks**

Name	Description	Benefits, Costs, and Limitations
MVC	<p>Model–view–controller (MVC) is a software architecture pattern, which separates the information manipulates by a program and depiction of user interaction with that program. This architecture includes model, view and controller, and following are the descriptions of each:</p> <p><b>Model:</b> When there are changes in value, a model sends notification to view and controller. This notification will let view to produce update and redraw, and controller to modify the view if its logic required.</p> <p><b>View:</b> View requests information from the model to get additional information to show views.</p> <p><b>Control:</b> A control can send commands to the model to update the model's state. It can also send commands to its associated view to change the view of the model.</p>	<p>Benefits:</p> <ol style="list-style-type: none"> <li>1. Separation of concerns: the separation of three components allows the reuse of the business logic across application.</li> <li>2. Independent development of information and depiction: developers only have to concern one part at a time.</li> </ol> <p>Cost:</p> <p>There is no specific cost required to use MVP architecture.</p> <p>Limitation:</p> <ol style="list-style-type: none"> <li>1. Increase of development complexity.</li> <li>2. Need multiple programming skill sets and knowledge.</li> <li>3. Inefficient of data access in views.</li> </ol>
Client-Server style	<p>Client-Server style is an architecture style. Clients send service request to the server, which performs the required functions and replies as needed with the requested information. Communication is initiated by the clients.</p> <p><b>Client:</b> User who need data from server.</p> <p><b>Server:</b> components that have API for user to request data.</p>	<p>Benefits:</p> <ol style="list-style-type: none"> <li>1. Centralization of computation and data at the server can make single server serve many clients at a time.</li> </ol> <p>Cost:</p> <p>There is no specific cost required to use client-server style.</p> <p>Limitation:</p> <ol style="list-style-type: none"> <li>1. The limitation of network bandwidth will influence the quality of service.</li> </ol>