System and Software Architecture Description (SSAD)

Pediatric Trauma Society Research Investigator Databank (PTS-RID)

#1

Kenda Albertson: IIV&V, Shaper

Georges Hatem: Project Manager, Lifecycle Planner

Mehrdad Mahdavi: Project Manager, Feasibility Engineer

Nicholas McCall: Operational Concept Engineer, Requirement Engineer

Junjian Wang: System Architect, Prototyper

March 31th 2013

Version History

Date	Author	Version	Changes made	Rationale
10/15/12	SA & JW	1.0	• First System and Software Architecture Description	• Based on the negotiations with the client so far
10/20/12	SA & JW	1.1	• Finish section 2.1.4 and 2.2 to complete the section 1 & 2	• Based on the negotiations with the client and info from winbook so far
10/21/12	SA & JW	1.2	• Fix some problems in the system behavior section	• Based on the evaluation from IIV&Ver
10/29/12	SA & JW	1.3	• Fix some problems in the section 1 & 2	• Based on the evaluation and comments from IIV & Ver and TA
11/04/12	SA & JW	1.4	• Fix some problems in the section 1& 2 (cont'd)	• Based on the evaluation from Professors
11/14/12	SA & JW	1.5	• Fix some problems in the system behavior part	• Based on the evaluation from TA and IIV & Ver.
11/26/12	SA & JW	1.6	• Complete all sections	• Based on the negotiation with the client
12/04/12	SA & JW	2.0	• Fix some problems in section 4	• Based on the evaluation of the whole team
12/10/12	SA & JW	2.1	• Fix some problems throughout the whole document	• Based on the evaluation of IIV & Ver and Professors
2/10/13	JW	3.0	• Fix some problems throughout the document	• Based on the evaluation of TA
2/19/13	JW	3.1	• Fix some problems throughout the document	• Based on the evaluation of TA
3/30/13	JW	4.0	• Fix some problems throughout the document	• Based on the evaluation of TA and feedback during the development

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the SSAD

The SSAD is the core document that is the result of design and analysis of the system and its requirements which is used by almost all the stakeholders, including the programmer, maintainer and clients. SSAD shows the whole structure of the system, the components used in the system and the interoperation between them. The developers use the SSAD as a reference to implement the system.

1.2 Status of the SSAD

This SSAD is of the version 4.0 which mainly fix some problems throughout the whole document based on the comments from TA and the feedback during the development. The document is for IOC1 Package.

2. System Analysis

2.1 System Analysis Overview

The Pediatric Trauma Society is a newly formed national organization dedicated to improving the outcomes of injured children. PTS aims to be a global leader in the field of pediatric trauma and injury prevention through optimal care guidelines, education, research, and advocacy. The PTS-RID system will pull key data from the external database PubMed and store all of them in a local database in order to enable Members to search it for forming collaborations. The database will act to facilitate research on a national basis and will be instrumental in improving pediatric trauma outcomes.

2.1.1 System Context

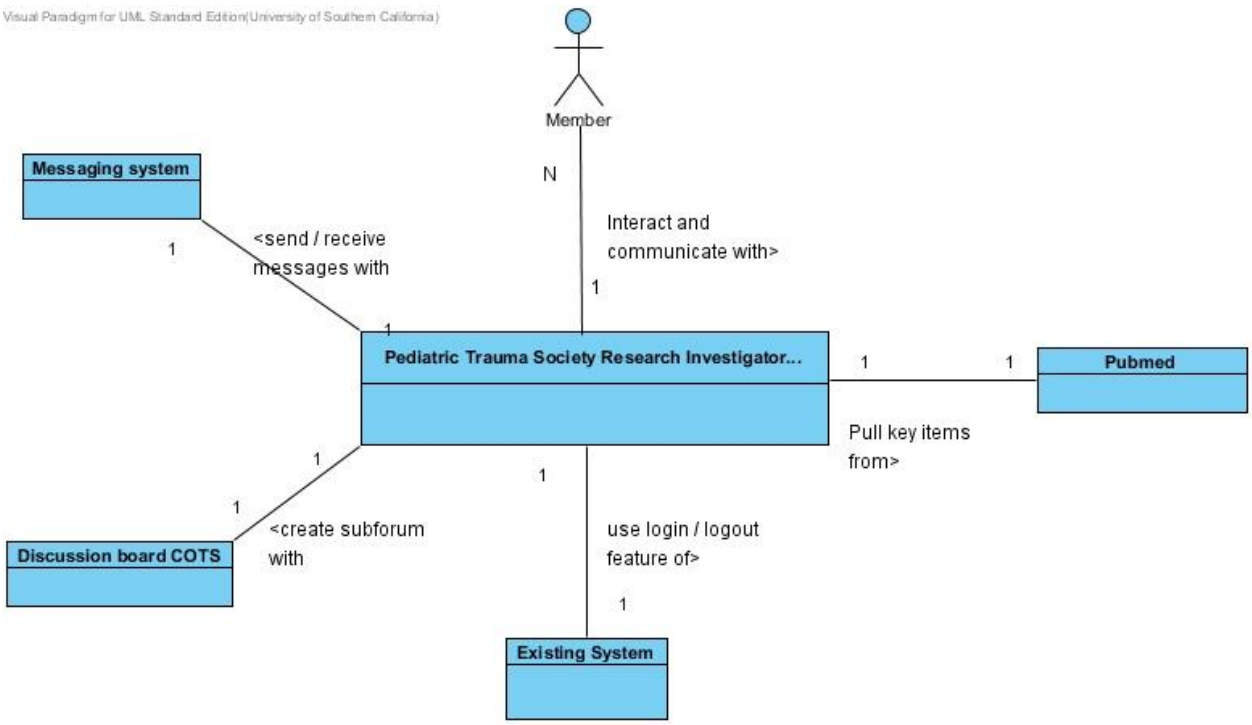

Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Pubmed	A free database providing academic topics and references in the domain of biomedicine and science.	● Expose API for accessing database items in the area of pediatrics, including journals, abstracts, MESH terms and so on.
Discussion board COTS	The COTS specializing at providing discussion-board-related functionality.	● Provision functionality of subforum creation, posts and comments
Members	The users which are the members of PTS	● Do searching ● Create Topic (Group) for discussion ● Send/Receive Messages ● View message history ● Post and comment ● Upload CV/Resume ● View graph in collaboration ● View articles
Existing System	A developed system of PTS	● Integrate and interact with PTS-RID, and provide login / logout feature to PTS-RID
Messaging System	A system that provides messaging functionality	● Provision sending/receiving message and viewing message history functionality

2.1.2 Artifacts & Information

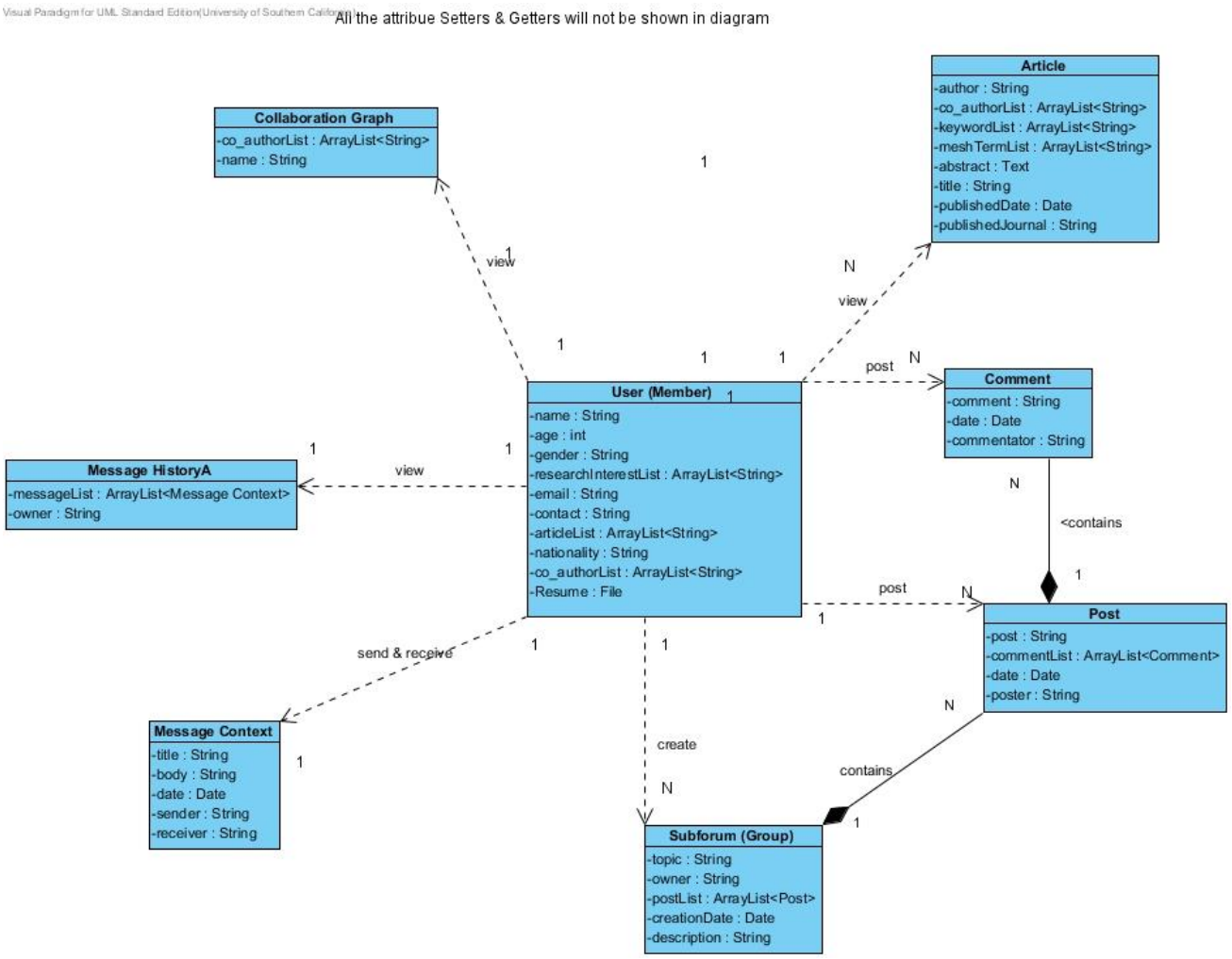

Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
User (Member)	Contains all information about the user that include personal information (name, gender, research area, etc.), the list of articles that he/she has ever published on Pubmed and the list of authors that he/she has ever collaborated with
Collaboration Graph	Illustrates the collaboration networks where the user has cooperate with other members in certain research area
Subforum (Group)	Contains all the public posts and comments by which members communicate with one another upon their research
Post	Post posted by a certain user
Comment	Comment made on a certain post
Message Context	Contains the message information and the context including sending / receiving date, sender and receiver
Message History	Contains the history of all the private messages that the user sends to/receives from other members
Article	Article that has been published in Pubmed by certain authors

2.1.3 Behavior

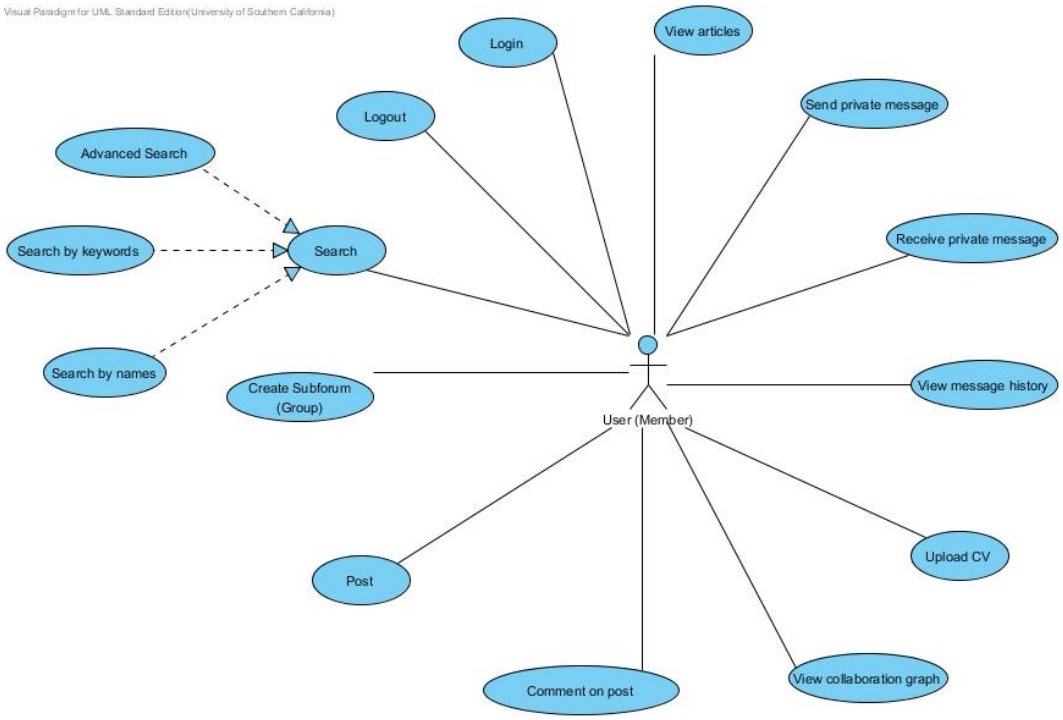

Figure 3: Process Diagram

2.1.3.1 Capability

2.1.3.1.1 Process Search

Table 8: Process Description

Identifier	UC-3: Search
Purpose	Allow users to search the information they want by keyword, MESH terms and names.
Requirements	WC_1527, WC_1500, WC_1493, WC_1492, WC_1491
Development Risks	Inability to handle 100 members doing concurrent search jobs. Response time not guaranteed.
Pre-conditions	The user has logged in, and is currently on the profile page
Post-conditions	A list of search result will show on the screen, or a message that nothing has been found

Table 9: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Input search item, and click Search Button	
2		Sends the request to PTS-RID back-end to do search processing, and find corresponding items
3		Display the list of search results

Table 10: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do search processing, and fail to find corresponding items
3		Display the message that no item matches your search criteria

2.1.3.1.2 Process Create Subforum (Group)

Table 11: Process Description

Identifier	UC-4: Create Subforum (Group)
Purpose	Create a subforum (group) for member discussion
Requirements	WC_1495
Development Risks	None
Pre-conditions	The user has logged in, and is currently on the profile page
Post-conditions	The group will successfully be created, or fail to be created

Table 12: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Configure group parameters, and click Create button	
2		Sends the request to PTS-RID back-end to do group creating processing, and group successfully created
3		Show the message that the group has been successfully created, and redirect the user to the group page

Table 13: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do group creating processing, and group cannot be created due to the name of the group not unique, or the information incorrectly being filled out
3		Display the message that group creation fails

2.1.3.1.3 Process Post

Table 14: Process Description

Identifier	UC-5: Post
Purpose	Post a message on the discussion board
Requirements	WC_1494
Development Risks	None
Pre-conditions	The user has logged in and the user is on any group where the user can post messages
Post-conditions	The post message will successfully be created, or fail to be created

Table 15: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Compose the message the user wants to post, and click Go button	
2		Sends the request to PTS-RID back-end to do user-post processing, and the post successfully created
3		Show that the message has been successfully posted

Table 16: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do user-post processing, and the post cannot be created due to some reasons like the word count outnumbers the limitation.
3		Show that the message has not been successfully posted.

2.1.3.1.4 Process Comment on post

Table 17: Process Description

Identifier	UC-6: Comment on post
Purpose	Comment on a post message on the discussion board
Requirements	WC_1494
Development Risks	None
Pre-conditions	The user has logged in, the user is currently on any group page where the user can post messages and there is any post message on which the user can comment
Post-conditions	The comment on post message will successfully be created, or fail to be created

Table 18: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Compose the comment the user wants to make on the post message, and click Go button	
2		Sends the request to PTS-RID back-end to do comment on user-post processing, and the comment is successfully created
3		Show that the comment on the post message has been successfully posted

Table 19: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do comment on user-post processing, and the comment cannot be created due to some reasons like the word count outnumbers the limitation.
		Show that the comment on the post message has not been successfully posted

2.1.3.1.5 Process View Collaboration graph

Table 20: Process Description

Identifier	UC-7: View collaboration graph
Purpose	View the collaboration relationship that the user has ever established with other members in a visualization way
Requirements	WC_1494
Development Risks	None
Pre-conditions	The user has logged in and the user is currently on the user profile page
Post-conditions	The collaboration graph is displayed, or show the message that no relationship found

Table 21: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click the View button	
2		Sends the request to PTS-RID back-end to do fetching collaboration relationship processing, and the relationship is successfully obtained
3		Build the graph in the client side and show the collaboration graph on the screen

Table 22: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do collaboration relationship processing, and find that the user has not established any relationship with others
3		Show the message that the user does not have any relationship with any of the members

2.1.3.1.6 Process Upload CV

Table 23: Process Description

Identifier	UC-8: Upload CV
Purpose	Upload CV in order to be viewed by other members
Requirements	WC_1498
Development Risks	None
Pre-conditions	The user has logged in, and the user is currently on the user profile page
Post-conditions	The CV is successfully uploaded, or fail to be uploaded

Table 24: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click the Browse button locate the CV on the local, and click Upload button to upload file	
2		Sends the request to PTS-RID back-end to do CV uploading processing, and CV is successfully uploaded.
3		Show the uploaded CV

2.1.3.1.7 Process Send private message

Table 26: Process Description

Identifier	UC-9: Send private message
Purpose	Send private message to other members in order to facilitate collaboration
Requirements	WC_1496
Development Risks	None
Pre-conditions	The user has logged in and the user is currently on the user profile page
Post-conditions	The message has been successfully sent and shown on the message list, or fail to be sent

Table 27: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Compose the message, designate the receiver and click Send button	
2		Sends the request to PTS-RID back-end to do sending private message processing, and the message is successfully sent.
3		Show the message being successfully sent and shown on the message list

Table 28: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do sending private message processing, and the message cannot be sent due to some reasons like the word count outnumbers the limitation..
		Show the message that the message has not been successfully sent

2.1.3.1.8 Process Receive private message

Table 29: Process Description

Identifier	UC-10: Receive private message
Purpose	Receive private message from other members in order to facilitate collaboration
Requirements	WC_1496
Development Risks	None
Pre-conditions	The user has logged in and the user is currently on the user profile page
Post-conditions	The received message is shown on the message list, or the message that no new message

Table 30: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click the Message Notification link	
2		Sends the request to PTS-RID back-end to do receiving private message processing, and find newly-received message
3		Show the received message

Table 31: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do receiving private message processing, and fail to find any newly-received message
3		Show the message of No New Message

2.1.3.1.9 Process View message history

Table 32: Process Description

Identifier	UC-11: View message history
Purpose	View the complete history of the message that the user has ever sent or received.
Requirements	WC_1497
Development Risks	None
Pre-conditions	The user has logged in and the user is currently on the user profile page
Post-conditions	The message history is shown on the screen, or the message that no message sent or received

Table 33: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click the View button	
2		Sends the request to PTS-RID back-end to do message history processing, and find the message history
3		Show the message history

Table 34: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do message history processing, and fail to find the message history for having not sent or received any message
3		Show No Messages in the history

2.1.3.1.10 Process View articles

Table 35: Process Description

Identifier	UC-12: View articles
Purpose	View the list articles that the user has ever published on Pubmed
Requirements	WC_1501
Development Risks	None
Pre-conditions	The user has logged in and the user is currently on the user profile page
Post-conditions	The list of articles is shown on the screen, or the message that no articles found

Table 36: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Click the View Articles button	
2		Sends the request to PTS-RID back-end to do article fetching processing, and the list of articles are successfully obtained
3		Show the list of articles

Table 37: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do article fetching processing, and the article cannot be found
		Show that no article exists

2.1.3.1.11 Process Advanced Search

Table 38: Process Description

Identifier	UC-13: Advanced search
Purpose	Facilitate users to search the articles they want by MESH terms.
Requirements	WC_1527, WC_1491
Development Risks	Inability to handle 100 members doing concurrent search jobs. Response time not guaranteed.
Pre-conditions	The user has logged in, and the user is currently on the user profile page
Post-conditions	A list of articles will show on the screen, or the message that no item is found

Table 39: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Select Country and State item, and click Search Button	
2		Sends the request to PTS-RID back-end to do advanced term search processing, and find the corresponding items
3		Display the list of articles

Table 40: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do advanced search processing, and fail to find any corresponding item
3		Display the message that no item matches your search criteria

2.1.3.1.12 Process Search by keywords

Table 41: Process Description

Identifier	UC-14: Search by keywords
Purpose	Facilitate users to search the articles they want by keywords.
Requirements	WC_1527, WC_1493
Development Risks	Inability to handle 100 members doing concurrent search jobs. Response time not guaranteed.
Pre-conditions	The user has logged in, and the user is currently on the user profile page
Post-conditions	A list of articles will show on the screen, or the message that no item is found

Table 42: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Input search keyword, and click Search Button	
2		Sends the request to PTS-RID back-end to do keyword search processing, and find the corresponding items
3		Display the list of articles

Table 43: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do keyword term search processing, and fail to find any corresponding item
3		Display the message that no item matches your search criteria

2.1.3.1.13 Process Search by names

Table 44: Process Description

Identifier	UC-15: Search by names
Purpose	Facilitate users to search the profiles of other members they want by names.
Requirements	WC_1527, WC_1492
Development Risks	Inability to handle 100 members doing concurrent search jobs. Response time not guaranteed.
Pre-conditions	The user has logged in, and the user is currently on the user profile page
Post-conditions	A list of user profiles will show on the screen, or the message that no item is found

Table 45: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Input search name, and click Search Button	
2		Sends the request to PTS-RID back-end to do name search processing, and find the corresponding items
3		Display the list of user profiles

Table 46: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Refer to typical course of action	
2		Sends the request to PTS-RID back-end to do name search processing, and fail to find any corresponding item
3		Display the message that no item matches your search criteria

2.1.4 Modes of Operation

The PTS-RID has only one mode, therefore there is nothing more to describe of the mode of operation.
Note that when the system does daily backup automatically, the server is still available to users.

2.2 System Analysis Rationale

Based on what we have analyzed from the system, we have figured out the following aspects that are less obvious or counter-intuitive:

- User: the users here stand for those who have already been a member of PTS the number of which is around 150. Note they do not need to sign up to access the PTS-RID pages because some of their profiles have been stored in the database once they joined in the society.
- Subforum: the concept represents the discussion board or the group where members can post and comment after it has been created.
- Pubmed: Pubmed is a free external database of government from where our system pulls items upon the domain of pediatrics.
- Login/Logout: Since our system will use the login/logout feature from existing PTS system, these two cases are removed from the document.

3. Technology-Independent Model

The section is skipped because all the key factors like particular hardware platforms, programming language, programming paradigm and so on have been specified.

4. Technology-Specific System Design

4.1 Design Overview

4.1.1 System Structure

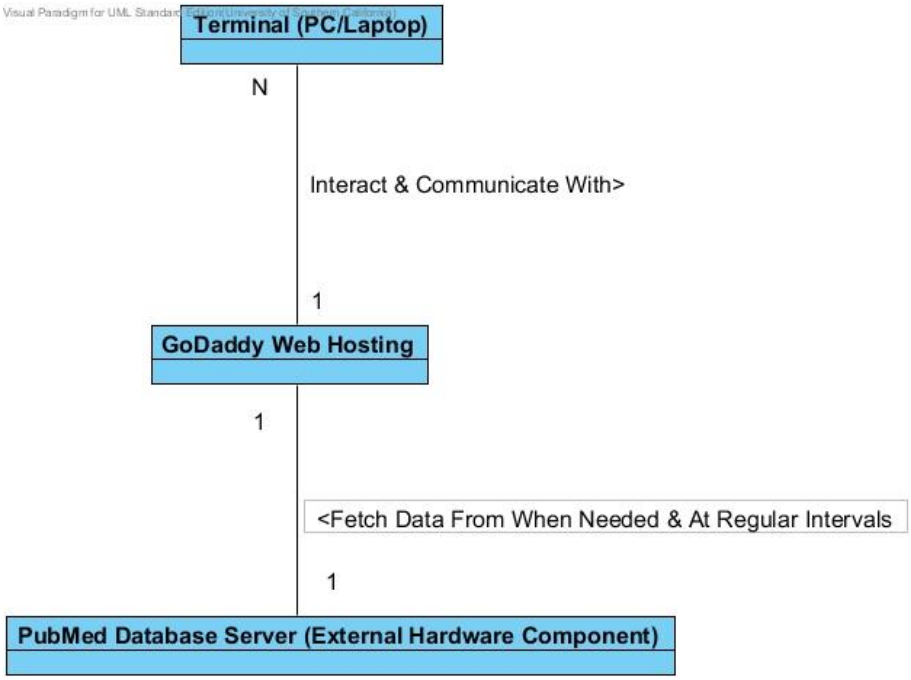

Figure 4: Hardware Component Class Diagram

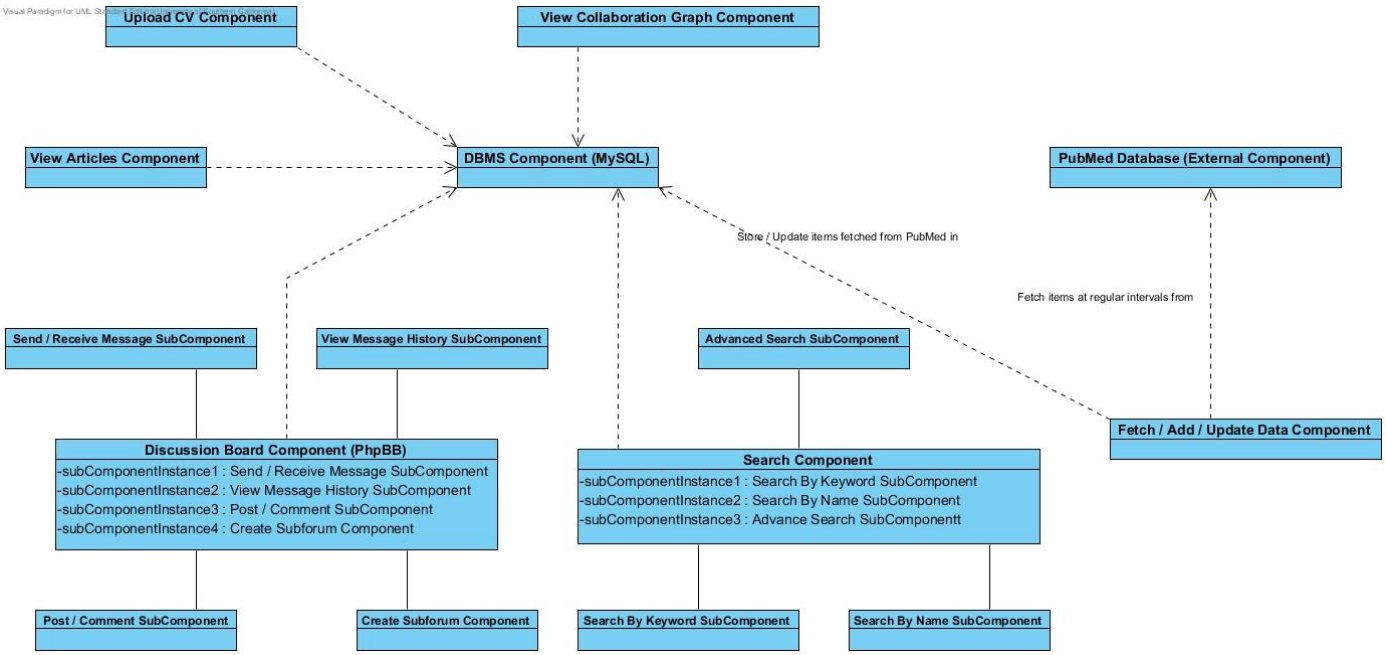

Figure 5: Software Component Class Diagram

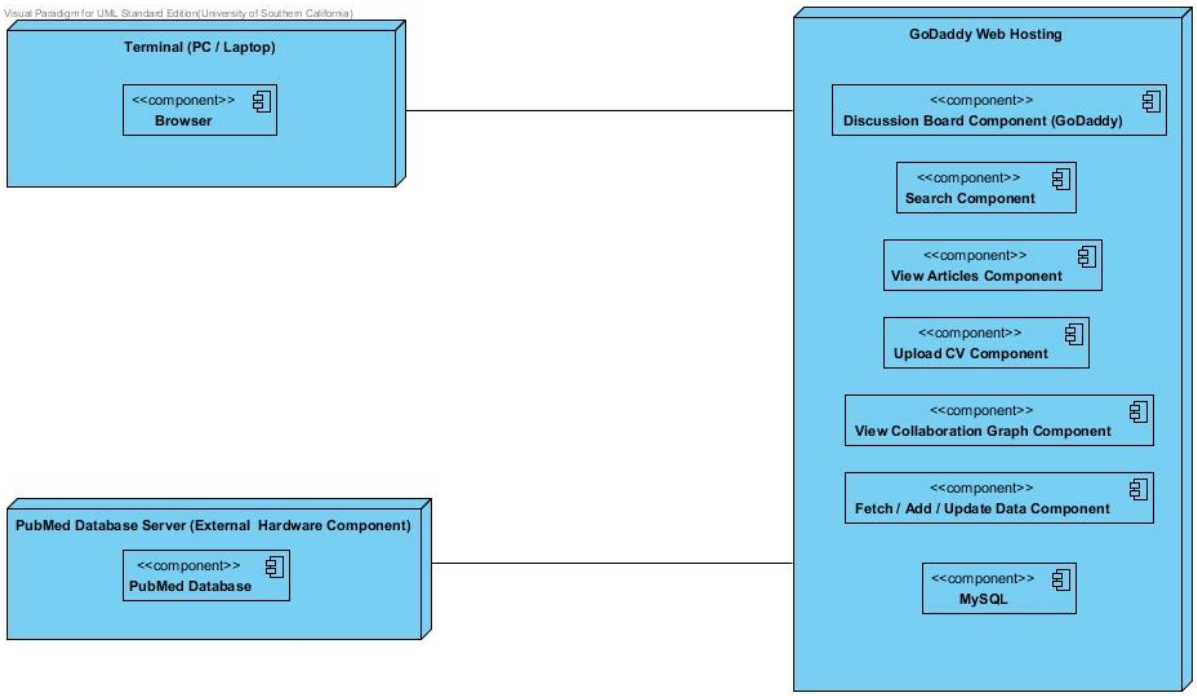

Figure 6: Deployment Diagram

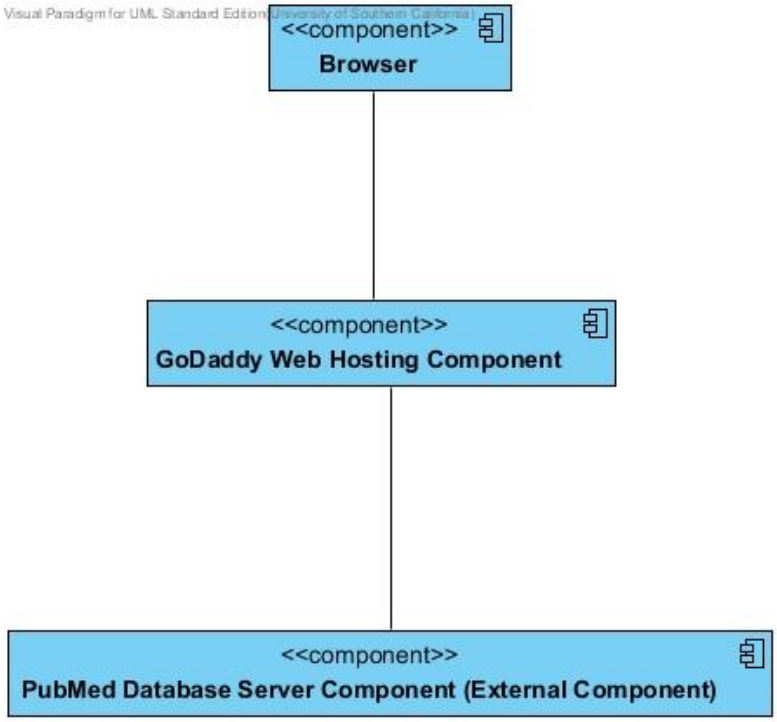

Figure 7: Supporting Software Component Class Diagram

Table 47: Hardware Component Description

Hardware Component	Description
Terminal (PC/Laptop)	Terminals are the PCs and laptops that PTS members use to access GoDaddy by means of requesting web pages via browser.
GoDaddy Web Hosting	GoDaddy is a paid web hosting that includes or stores (in a static way): ● MySQL database ● Web pages & Scripts ● Files (resumes & articles of the PTS members) In addition, it contain the functionality such as (in a dynamic way): ● Do all the logic processing corresponding to the requests from clients (terminals). ● Fetch and update existing data at regular intervals (once every night at the moment).
Pubmed Database Server (External Component)	Pubmed Database Server is the external component that GoDaddy fetches data from, which here is just used to show the relationship between the components involved in our system.

Table 48: Software Component Description

Software Component	Description
Discussion Board Component (PhpBB)	PhpBB is a paid discussion board COTS that includes the following subcomponents our system needs: ● Create Subforum SubComponent: responsible for creating subforum for members to share information ● Post / Comment SubComponent: responsible for handling members' post and comment request that facilitate them with communication and interaction. ● Send / Receive Message SubComponent: responsible for processing the functionality of sending and receiving private messages between members. ● View Message History Component: responsible for showing the message history of a member with others.
Search Component	Search Component is the one that handling the search requests from members. There are three kinds of searching subcomponents: ● Search By Keyword SubComponent: responsible for handling keyword-search requests from members. If the items corresponding to the keyword do not exist in the local database, return null. ● Search By Names SubComponent: responsible for handling author-name-search requests from members. If the items corresponding to the name do not exist in the local database, return null. ● Advanced Search SubComponent: responsible for handling advanced-search requests from members. Currently advanced search includes searching members by Country and State. If the items corresponding to the country and state do not exist in the local database, return null.
View Articles Component	View Articles Components simply handles the request to view the list of articles a member has published before in Pubmed.
Upload CV Component	Upload CV Component is responsible for the member to upload their resumes.
View Collaboration Graph Component	View Collaboration Graph Component shows the relationship graph between the member and the others that have been ever collaborated with.
DBMS Component (MySQL)	MySQL is the database (currently we use the free version of it and may transfer into the paid version in the future) that store all the items fetched from Pubmed.
Fetch / Add / Update Data Component	Fetch / Add / Update Data Component is the one that fetches key items from Pubmed at regular intervals and then store them or update the corresponding data in MySQL in the back end. It is the only component in our system that does not directly interact with the users.

Pubmed Database (External Component)	Pubmed Database is the external software component that provides the key items that our system needs.

Table 49: Supporting Software Component Description

Support Software Component	Description
Browser	Browser is the most common component that resides on the terminal that clients use to request web pages from GoDaddy and to interact with it
GoDaddy Web Hosting Component	GoDaddy Web Hosting Component includes MySQL, web pages & scripts and files that are fetched and stored by PTS Application Server Component. In addition, it is where all the logic components reside except for MySQL and Pubmed. All the computation processing is done here.
Pubmed Database Server Component (External Component)	Pubmed Database Server Component is an extern component that provides all the data needed by our system.

4.1.2 Design Classes

4.1.2.1 Class Diagram

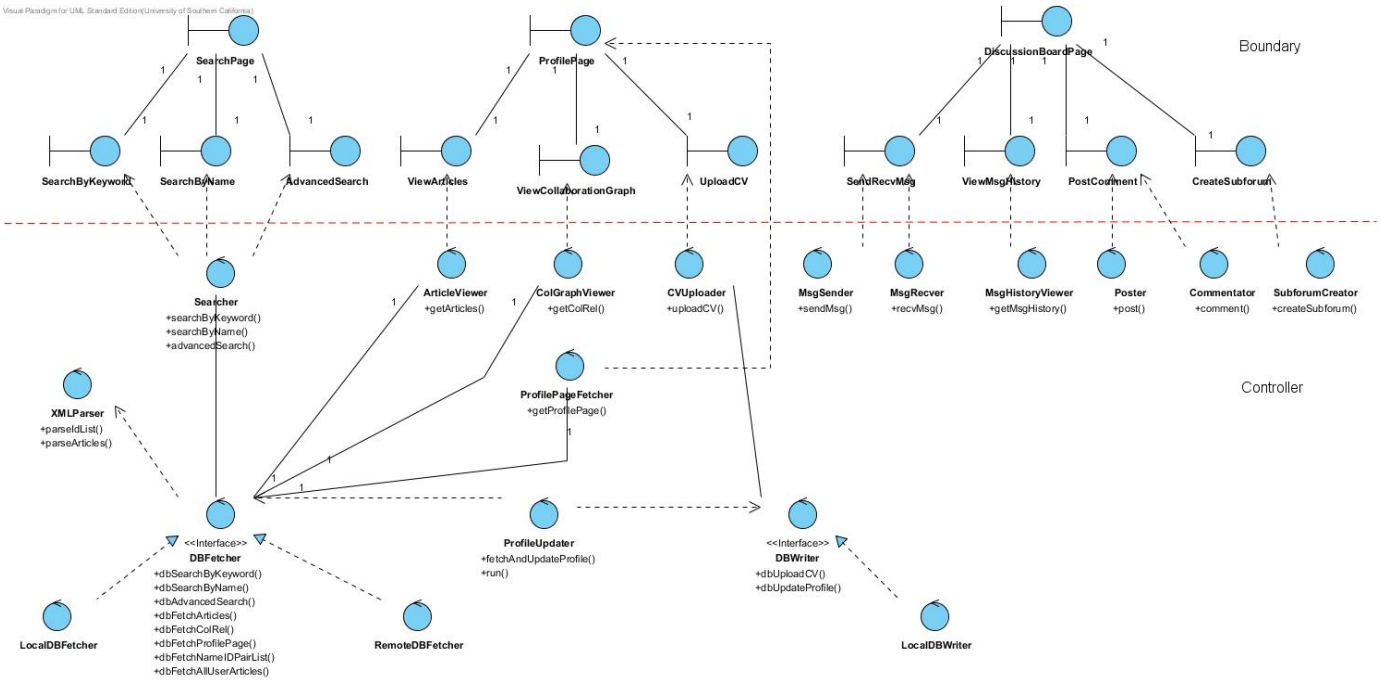

Figure 8: Boundary & Control Class Diagram

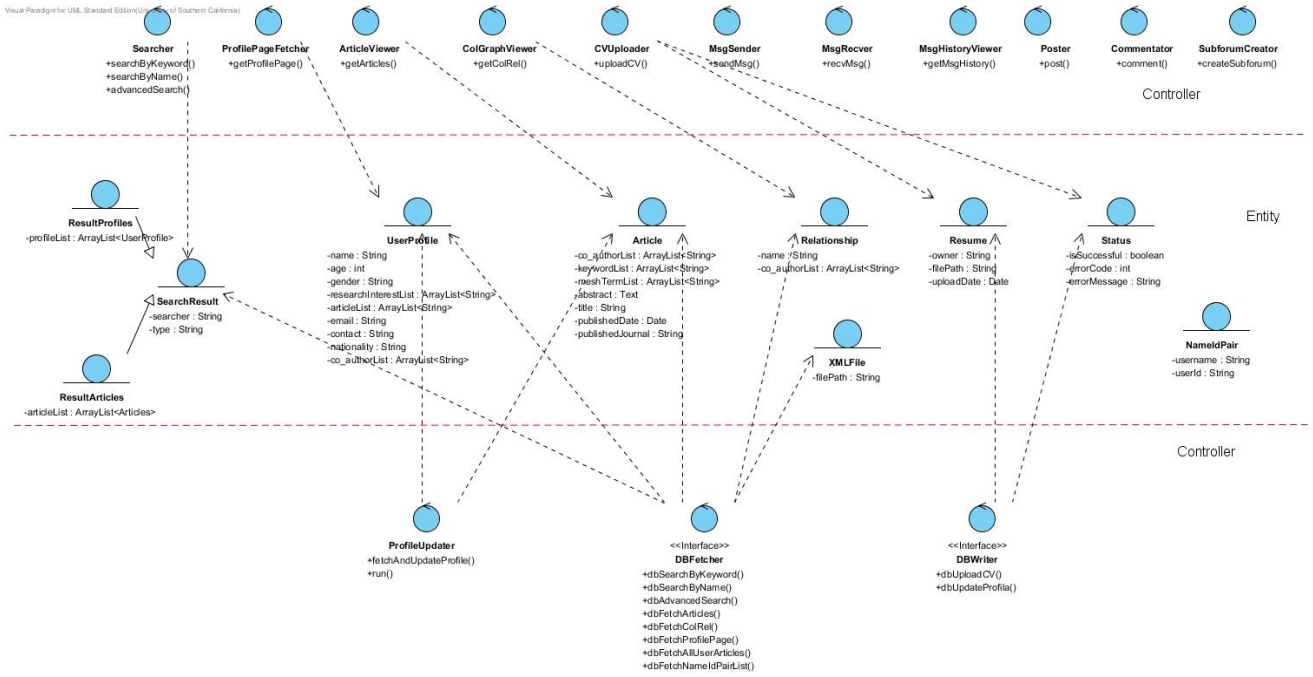

Figure 9: Control & Entity Class Diagram

Table 50: Boundary Class Description

Class	Type	Description
SearchPage	Boundary	Search page containing search by keyword area, search by name area and advanced search area
SearchByKeyword	Boundary	Search by keyword area
SearchByName	Boundary	Search by name area
AdvancedSearch	Boundary	Advanced search area containing country and state dropdown menu selection
ProfilePage	Boundary	User profile page containing all the user information, including view articles area, view collaboration graph area and upload CV area
ViewArticles	Boundary	View article area that the user can view the articles of himself
ViewCollaborationGraph	Boundary	View collaboration area that the user can view the collaboration graph based on the map
UploadCV	Boundary	Upload CV area that allows the user to upload their CV
DiscussionBoardPage	Boundary	Discussion board page containing send/receive message area, view message history area, post/comment area and create subforum area
SendRecvMsg	Boundary	Send/receive message area that allows the user to send/receive message from/to the other PTS members
ViewMsgHistory	Boundary	View message history area containing all the messages the user has ever sent or received
PostComment	Boundary	Post/comment area that allows the user to post or comment on the post
CreateSubforum	Boundary	Create subforum area that allows the user to create a subforum with certain topic

Table 51: Control Class Description

Class	Type	Description
Searcher	Control	The class responsible for searching which contains the following methods: ● SearchByKeyword (userId: String, keyword: String): SearchResult ● SearchByName (userId: String, name: String): SearchResult ● advancedSearch (userId: String, country: String, state: String): SearchResult Package: pts.ptsrid.component
ArticleViewer	Control	The class responsible for fetching articles which contains the following methods: ● getArticles (userId: String): ArrayList<Article> Package: pts.ptsrid.component
ColGraphViewer	Control	The class responsible for fetching collaboration relationship which contains the following methods: ● getColRel (userId: String): Relationship Package: pts.ptsrid.component
CVUploader	Control	The class responsible for uploading CV which contains the following methods: ● uploadCV (userId: String, filePath: String): Status Package: pts.ptsrid.component
ProfilePageFetcher	Control	The class responsible for fetching user profile which contains the following methods: ● getProfilePage (userId: String): UserProfile Package: pts.ptsrid.component
MsgSender	Control	The class responsible for sending message which contains the following methods: ● sendMsg () Package: pts.ptsrid.component
MsgRecver	Control	The class responsible for receiving message which contains the following methods: ● recvMsg() Package: pts.ptsrid.component
MsgHistoryViewer	Control	The class responsible for fetching message history which contains the following methods: ● getMsgHistory() Package: pts.ptsrid.component

Poster	Control	The class responsible for posting which contains the following methods: ● post () Package: pts.ptsrid.component
Commentator	Control	The class responsible for commenting on the post which contains the following methods: ● Comment () Package: pts.ptsrid.component
SubforumCreator	Control	The class responsible for creating subforum which contains the following methods: ● createSubforum() Package: pts.ptsrid.component
DBFetcher	Control Interface	The interface responsible for all the fetching job from database that serves as an auxiliary component to those classes whose task is to read data from database which contains the following methods: ● dbSearchByKeyword (userId: String, keyword: String): SearchResult ● dbSearchByName (userId: String, name: String): SearchResult ● dbAdvancedSearch (userId: String, country: String, state: String): SearchResult ● dbFetchArticles (userId: String, username: String): ArrayList<Article> ● dbFetchColRel (userId: String): Relationship ● dbFetchProfilePage (userId: String): UserProfile ● dbFetchNameIdPairList (): ArrayList<NameIdPair> ● dbFetchAllUserArticles (usernameList: ArrayList<String>): ArrayList<ArrayList<Article>> Package: pts.ptsrid.db
LocalDBFetcher	Control	The class responsible for fetching data from the local database which should implement the following methods (in terms of other methods that need not be implemented, just remain them empty in the method body): ● dbSearchByKeyword () ● dbSearchByName () ● dbAdvancedSearch () ● dbFetchArticles () ● dbFetchColRel ()

		● dbFetchProfilePage () ● dbFetchNameIdPairList () Package: pts.ptsrid.db.impl
RemoteDBFetcher	Control	The class responsible for fetching data from the remote database (Pubmed) when the local database does not have the data corresponding to the request which should implement the following methods (in terms of other methods that need not be implemented, just remain them empty in the method body): ● dbFetchAllUserArticles() Package: pts.ptsrid.db.impl
DBWriter	Control Interface	The interface responsible for all the writing job into database that serves as an auxiliary component to those classes whose task is to write data into database which contains the following methods: ● dbUploadCV (userId: String, filePath: String): Status ● dbUpdateProfile (articles: ArrayList<Article>): Status Package: pts.ptsrid.db
LocalDBWriter	Control	The class responsible for writing data into the local database which should implement the following methods (in terms of other methods that need not be implemented, just remain them empty in the method body): ● dbUploadCV() ● dbUpdateProfile () Package: pts.ptsrid.db.imp
ProfileUpdater	Control	The class responsible for fetching user profiles (including their articles) from the remote server (Pubmed) and update all the existing data in the local database which contains the following methods: ● run (): void ● fetchAndUpdateProfile (): Status Note that the interval is once every night Package: pts.ptsrid.component

Table 52: Entity Class Description

Class	Type	Description
SearchResult	Entity	The entity representing the result of search Package: pts.ptsrid.datamodel
ResultProfiles	Entity	Derived from SearchResult. Return value of the search by name & advanced search Package: pts.ptsrid.datamodel
ResultArticles	Entity	Derived from SearchResult. Return value of the search by keyword Package: pts.ptsrid.datamodel
UserProfile	Entity	The entity that contains all the information of the user Package: pts.ptsrid.datamodel
Article	Entity	The entity representing the article that the user has ever published Package: pts.ptsrid.datamodel
Relationship	Entity	The entity representing the collaboration relationship between the user and the other members Package: pts.ptsrid.datamodel
Resume	Entity	The entity representing the CV of the user Package: pts.ptsrid.datamodel
Status	Entity	The entity representing the status of the post request from the user Package: pts.ptsrid.datamodel
NameIdPair	Entity	The entity containing the user ID and username Package: pts.ptsrid.datamodel

4.1.3 Process Realization

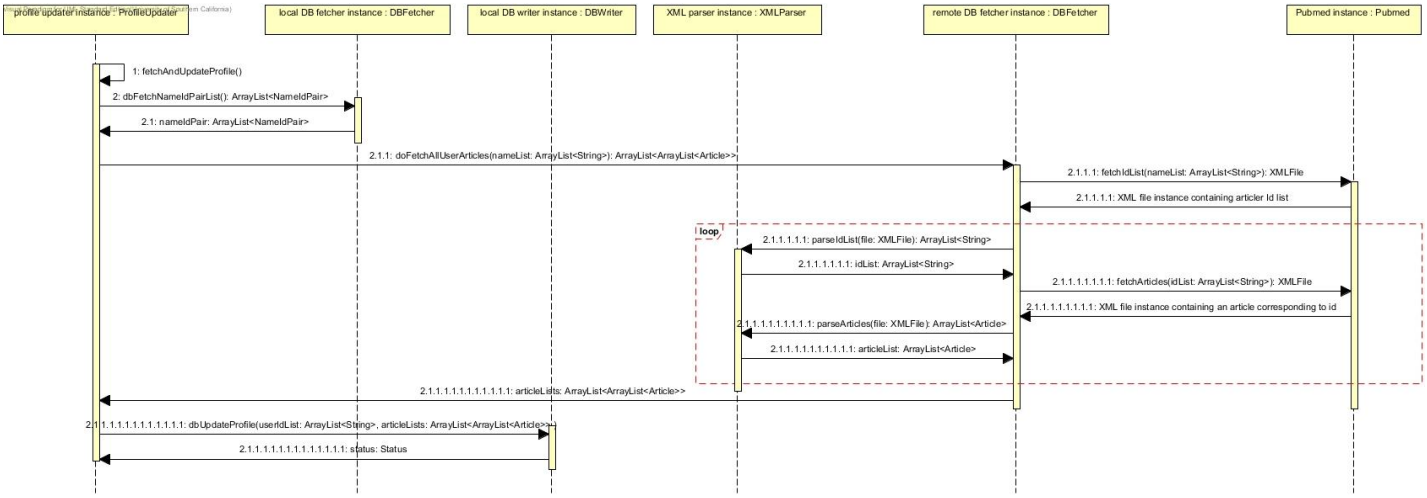

Figure 10: Process Realization Diagram

4.1.4 E-R Diagram (physical level)

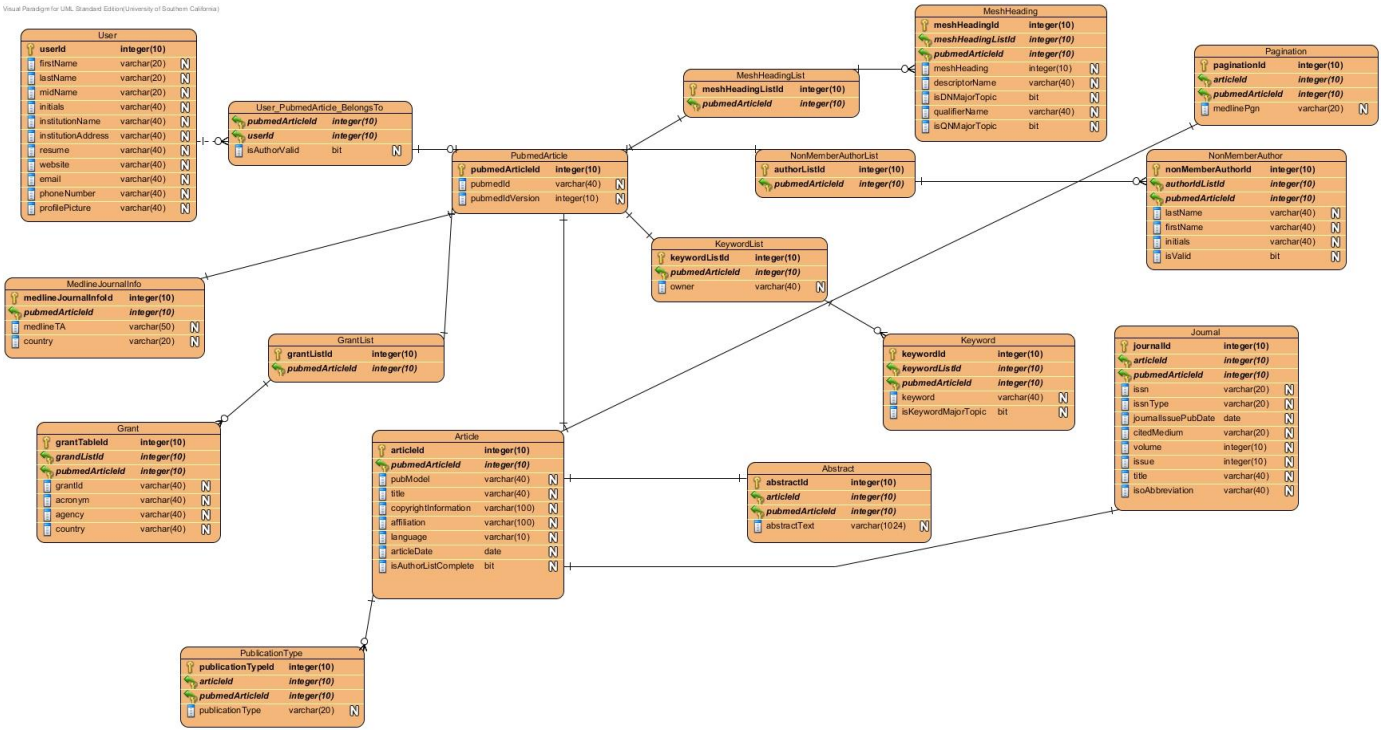

Figure 11: ER Diagram

4.2 Design Rationale

Our system PTS-RID is of a typical B/S framework. So we adopt MVC pattern for our project: M corresponds to the Entity Class, V corresponds to the Boundary Class and C corresponds to the Control Class.

In addition to that, regarding the COTS we choose to use, there are two candidates available: myBB and phpBB. Here is the comparison table: (O stands for good, + stands for very good. In addition there is no note here just because of its informality)

Table 33 Comparison table

Criteria	MyBB	phpBB
Documentation	O	+
Modding process	+	O
Additional features	+	O
Security history	O	+
Admin UI simplicity	+	O
Comprehensive settings	O	+
Adding Styles/Themes	+	O

Detailed Comparison:
http://www.forum-software.org/forum-comparator/mybb-vs-phpbb3

According to our client and the table above, finally we determine phpBB as our discussion board COTS. The reason for using it is that our client cares more about the security problem, and we developers tend to choose the one with more detailed user manual for ease of development.

What's more, the reason to choose GoDaddy as the web hosting is:

- GoDaddy has provided large space for storage of whether the database or the files.
- PTS has deployed there existing system on GoDaddy for a long time without any problem, so the stability and the reliability could be guaranteed.
- Some features like login and logout have been already scratched by PTS. Therefore staying on the same server with PTS existing system facilitates integration.

5. Architectural Styles, Patterns and Frameworks

Table 54: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations
B/S	Brower/server architecture pattern	Benefit: client side (browser) do not need to do much Costs: not much Limitations: Server may undertake too much workload