

Campus Talk - Project Part 2

Team: Aadish Gupta
Pallavi Madasu
Yogitha Mahadasu

Title: Campus Talk

Project Summary: Platform for all the campus students to keep up to date with latest events happening across the campus. People can find other peers across the university with similar interests, say finding a group of people who are interested in Design Patterns or finding people who want to go for skiing a particular weekend. Users can create forums to initiate discussions and peers can subscribe to the forum to participate in ongoing discussions. Users can interact with each other by posting in the forums or by replying to existing posts. Users could also unsubscribe if they are no longer interested in following a group.

Project Requirements:

1. Business Requirements:

Business Requirements			
ID	Requirement	Topic Area	User
BR-001	Login email must be 'colorado.edu'	Authentication	All

2. User Requirements:

User Requirements			
ID	Requirement	Topic Area	User
UR-001	User Signs Up at the application	Authentication	All
UR-002	User Logs into the application	Authentication	All
UR-003	User can view his/her profile	Profile	All
UR-004	User can modify his profile page	Profile	All
UR-005	User can invite friends through email	Social	All

UR-006	User can view the events/posts on the news feed	Feed	All
UR-007	User can search for interested forums/ people with similar interests	Search	All
UR-008	User can click on the displayed search results and can look at the forum/person profile.	Search	All
UR-009	User can subscribe to the forum	Forum	All
UR-010	User can un-subscribe from the forum	Forum	All
UR-011	User can create a forum	Forum	All
UR-012	User can create a new forum post.	Forum	All
UR-013	User can reply to the forum post	Forum	All
UR-014	User can logout of the application	Authentication	All

3. Functional Requirements:

Functional Requirements		
ID	Requirement	Topic Area
FR-001	Validate the inputs	Data Entry
FR-002	Hash the password	Data Entry
FR-003	Authenticate the user	Authentication
FR-004	Send an email to the invited user with a link to signup page.	Social
FR-005	Updating the events from various sources to the database every midnight	Feed
FR-006	Retrieve the campus events information from the DB, sort them chronologically and display them on the news feed.	Feed
FR-007	Retrieve the user subscribed forums information from the DB, sort them according to the timestamp and	Feed

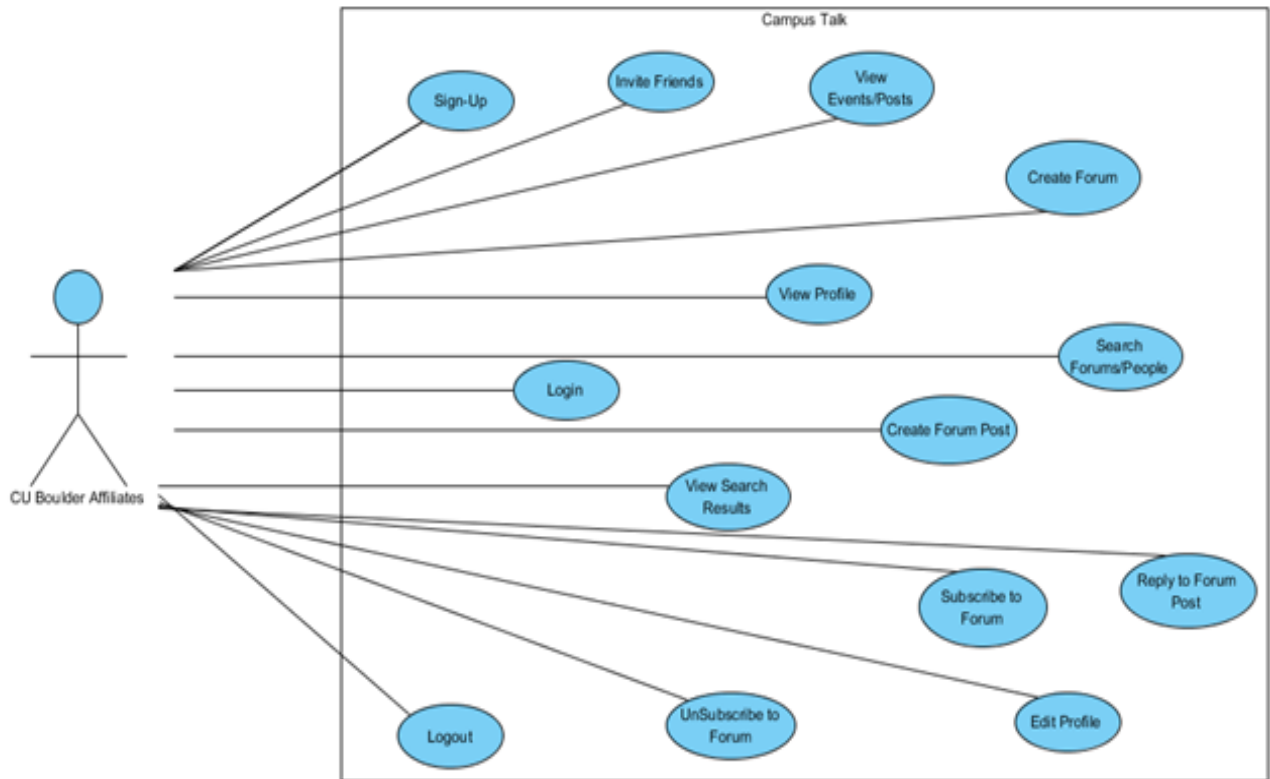
	display them on the activity feed.	
FR-008	Maintain user session throughout	User Session

4. Non-Functional Requirements:

Non-Functional Requirements			
ID	Requirement	Topic Area	User
NFR-01	User login shouldn't take more than 5 seconds after providing valid credentials	Latency	All
NFR-02	Retrieval of search results shouldn't take more than 7 seconds.	Latency	All
NFR-03	Responsive pages for all devices (different screen sizes and platforms)	Cross Platform	All

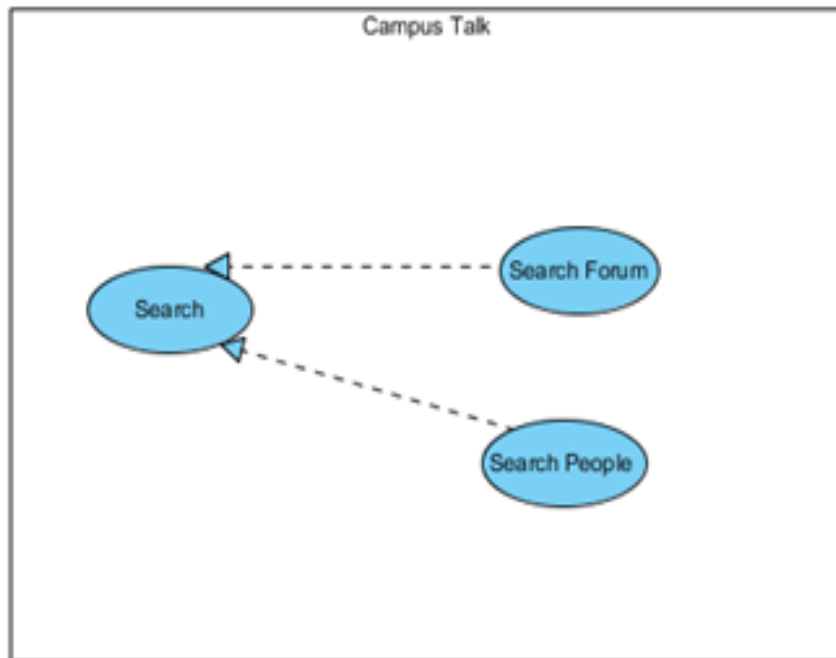
Use Cases

- **Actors:** CU Boulder Affiliates
- **Use Case Overview**

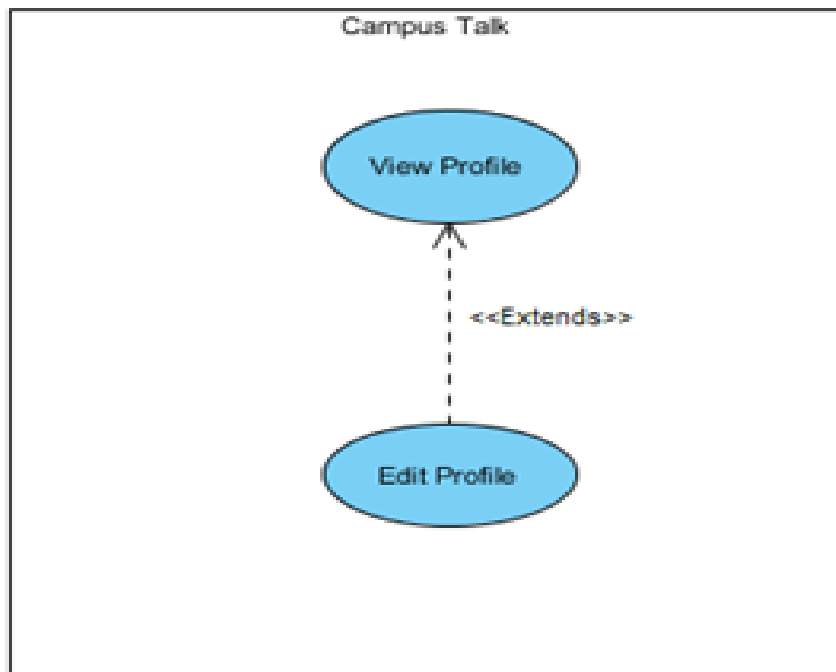


- Sub-diagrams

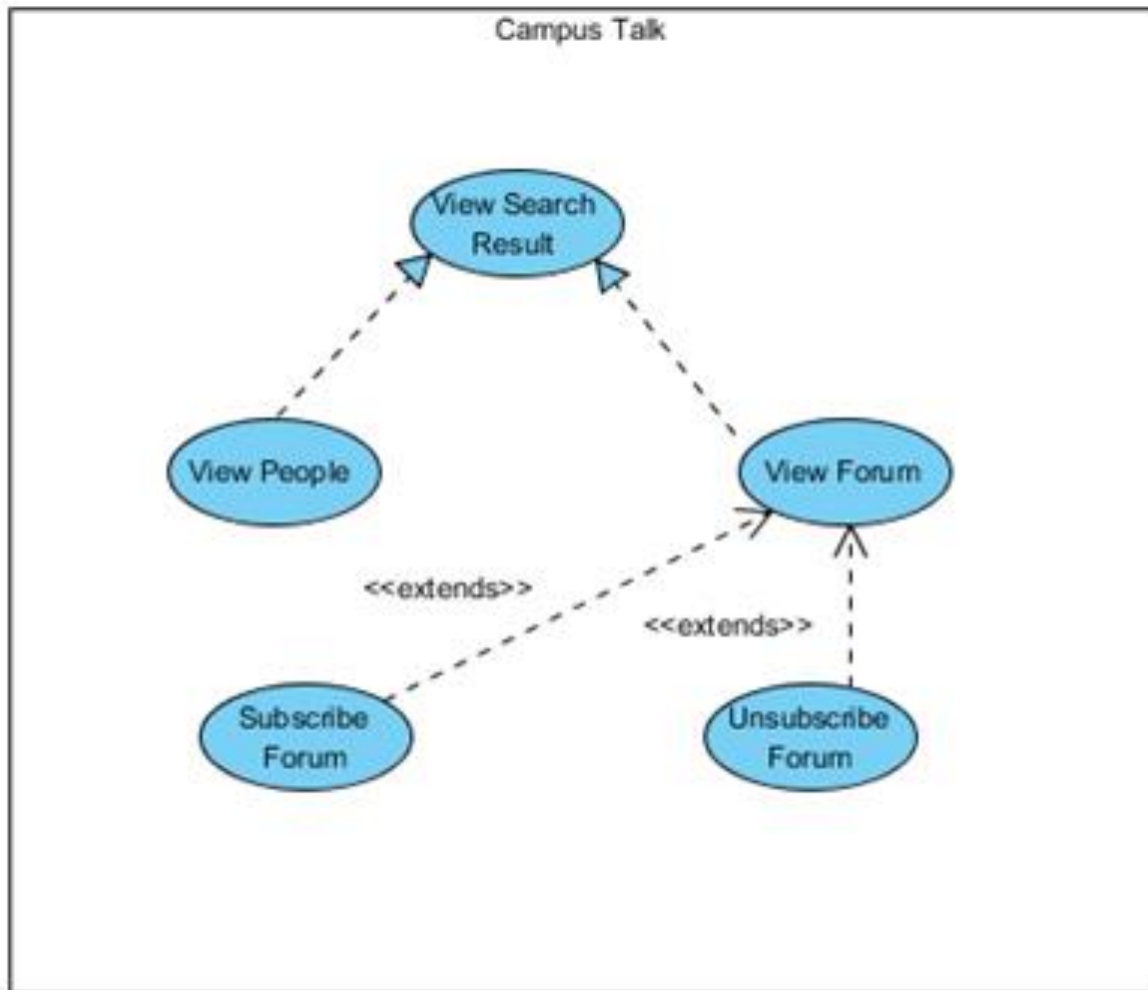
Sub Diagram 1



Sub Diagram 2



Sub Diagram 3



- Use Case Documents

Use Case ID:	UR-001
Use Case Name:	User signs Up at the application
Description:	User signs up at the web portal by providing details like his email ID, name, department, interests.

Actors:	CU Boulder Affiliate		
Pre-conditions:	User should have a colorado.edu email ID		
Post-conditions:	A profile has been created for the CU Boulder Affiliate (User).		
Frequency of Use:	Only once – the first time a cu boulder affiliate wants to create an account.		
Flow of Events:		Actor Action	System Response
	1	Provide his/her details by filling the sign-up form.	Validate the input and display error messages appropriately.
	2	Click on Sign-up button.	Store the details in DB associate the profile with the tags. Redirect the user to the homepage.
Variations:			
Exceptions:	Incorrect format of input results in error messages during sign-up		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-002
Use Case Name:	User logs into the application
Description:	User logs in the application by providing email ID and password.

Actors:	CU Boulder Affiliate		
Pre-conditions:	User has a profile created.		
Post-conditions:	User's home page is displayed after successful login.		
Frequency of Use:	Daily		
Flow of Events:		Actor Action	System Response
	1	Provide his/her login details.	Validate the inputs and display error messages appropriately
	2	Click on Log-in button.	Authenticate the user and display homepage on success.
Variations:			
Exceptions:	Incorrect format/invalid credentials results in error messages during login.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-003
Use Case Name:	User can view his/her profile
Description:	User can view his/her profile by clicking on the Profile button

Actors:	CU Boulder Affiliate		
Pre-conditions:	User must be logged in		
Post-conditions:	User's profile page is displayed.		
Frequency of Use:	Frequently		
Flow of Events:		Actor Action	System Response
	1	Click on Profile button.	User's profile page is displayed by dynamically retrieving details from DB
Variations:			
Exceptions:			
Notes and Issues:			
Developer Notes:	For every navigation in the application, session validation should happen.		

Use Case ID:	UR - 004
Use Case Name:	User can modify his profile page.
Description:	User can navigate to the profile page and edit his/her details.

Actors:	CU Boulder Affiliate (student/faculty)		
Pre-conditions:	User must be logged in		
Post-conditions:	User views his/her updated profile details.		
Frequency of Use:	Often		
Flow of Events:		Actor Action	System Response
	1	Click on Profile button in Home Page	User will be taken to his/her profile page.
	2	Click on Edit Profile.	Existing profile details are displayed in editable mode.
	3	Update the details	Validate the inputs and show corresponding error messages
	4	Click on Save button	Update profile details in the DB and render the profile page back in non-editable mode
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR – 005
Use Case Name:	User can invite friends through email
Description:	User can send invite to friends to sign-up for the application in the form of an email

Actors:	CU Boulder Affiliate (student/faculty)		
Pre-conditions:	User must be logged in.		
Post-conditions:	Invited friends should have received an invitation email.		
Frequency of Use:	Often by students/faculty		
Flow of Events:		Actor Action	System Response
	1	Click on Invite Friends button	User will be taken to a page where friends can be invited.
	2	Provide email ids of friends to be invited	Validate provided e-mail id's for Colorado.edu domain
	3	Click on Send Invite button	Send e-mail to the corresponding user address and display a message.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR – 006
Use Case Name:	User can view the events/posts on the news feed
Description:	User can navigate to the Home page and view the events/posts on the news feed

Actors:	CU Boulder Affiliate (student/faculty)		
Pre-conditions:	User must be logged in.		
Post-conditions:	User views events or posts on his/her timeline or news feed		
Frequency of Use:	Often		
Flow of Events:		Actor Action	System Response
	1	Click on Home button from anywhere in the application	Fetches the events and posts based on user preferences from DB and display on the user's homepage
Variations:	Login to the portal and the user will be landed in home page where all the events and posts can be viewed		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR - 007
Use Case Name:	User can search for interested people with similar interests.
Description:	User can look up for forums or people of his/her interest

Actors:	CU Boulder Affiliate (student/faculty)		
Pre-conditions:	User must be logged in.		
Post-conditions:	User views the search results based on the search criteria provided		
Frequency of Use:	Often		
Flow of Events:		Actor Action	System Response
	1	Click on Search button in Home Page	User will be taken to a page where search criteria will be shown
	2	Provide search criteria and click on search people or search group button	Search Results will be fetched from DB based on chosen criteria and displayed to the User using associated tags.
Variations:	User can search for interested Forums and follows the same process		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-008
Use Case Name:	User can click on the displayed search results and can look at the forum/person profile.
Description:	Once the search results are displayed user can then click on the given search result and will be redirected accordingly

Actors:	CU Boulder Affiliate (User)		
Pre-conditions:	User must be logged in and search results are displayed		
Post-conditions:	User is redirected to the clicked result which points to a forum page or a user profile page		
Frequency of Use:	Often		
Flow of Events:		Actor Action	System Response
	1	Clicks on the displayed search result	Redirects the user to the corresponding page linked with clicked search result
Variations:	The links generated can be stored by user for future reference and can be directly used to open the particular forum or person profile		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-009
Use Case Name:	User can subscribe to the forum
Description:	Once the user is on a particular forum page he/she can subscribe to the forum to receive the updates from the forum in his/her feed

Actors:	CU Boulder Affiliate (User)		
Pre-conditions:	User must be logged in and on the forum page		
Post-conditions:	User after clicking subscribe button sees unsubscribe button		
Frequency of Use:	Often		
Flow of Events:		Actor Action	System Response
	1		Renders the page and shows subscribe button if the user has already not subscribed the forum
	2	Click on the Subscribe button	Create a mapping between the user and the forum at the database level and update subscribe button to unsubscribe button
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-010
Use Case Name:	User can un-subscribe from the forum
Description:	User can un-subscribe from the forum to stop receiving the forum update in his/her feed

Actors:	CU Boulder Affiliate (User)		
Pre-conditions:	User must be logged in, on the forum page and subscribed to the forum		
Post-conditions:	User after clicking un-subscribe button sees subscribe button		
Frequency of Use:	Rarely		
Flow of Events:		Actor Action	System Response
	1		Renders the page and shows un-subscribe button if the user is already subscribed to the forum
	2	Clicks on the un- subscribe button	Removes the mapping between the user and the forum at the database level and update un-subscribe button to subscribe button on the page
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-011
Use Case Name:	User can create a Forum
Description:	User can start a new forum for people to have discussion on that forum topic

Actors:	CU Boulder Affiliate (User)		
Pre-conditions:	User must be logged in and on the home page		
Post-conditions:	New Forum is created and ready for discussion among users		
Frequency of Use:	Frequently		
Flow of Events:		Actor Action	System Response
	1	Clicks on Create Forum button	Redirects to create forum page
	2	Enters the forum title and tags which describe the forum.	Validates the input and display error message on the page
	3	Clicks on submit button	Check if the newly added forum satisfies the unique constraint (Title and Tags). If yes add the forum entry in the database, subscribe the user to that forum. Add associated tags to the Forum in the database. Redirect to newly created forum else display error message to the user and redirect to existing forum page.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-012
Use Case Name:	User can create a new forum post.
Description:	User can create a forum post whenever he wants to start a discussion/share some information.

Actors:	CU Boulder Affiliate		
Pre-conditions:	User must be logged in and subscribed to the forum in which he/she wants to create a new post.		
Post-conditions:	Forum post will be visible to all other users who have subscribed to the forum.		
Frequency of Use:	Frequently		
Flow of Events:		Actor Action	System Response
	1	Navigate to the forum in which he/she wants to post.	Redirect to the desired forum page
	2	Click on Create Post button.	A text box will be displayed to the user
	3	Enter the information and click on Post button.	An entry for the post is created in the DB and displayed on forum page.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-013
Use Case Name:	User can reply to the forum post
Description:	User can reply to a forum post to continue discussion/ask questions

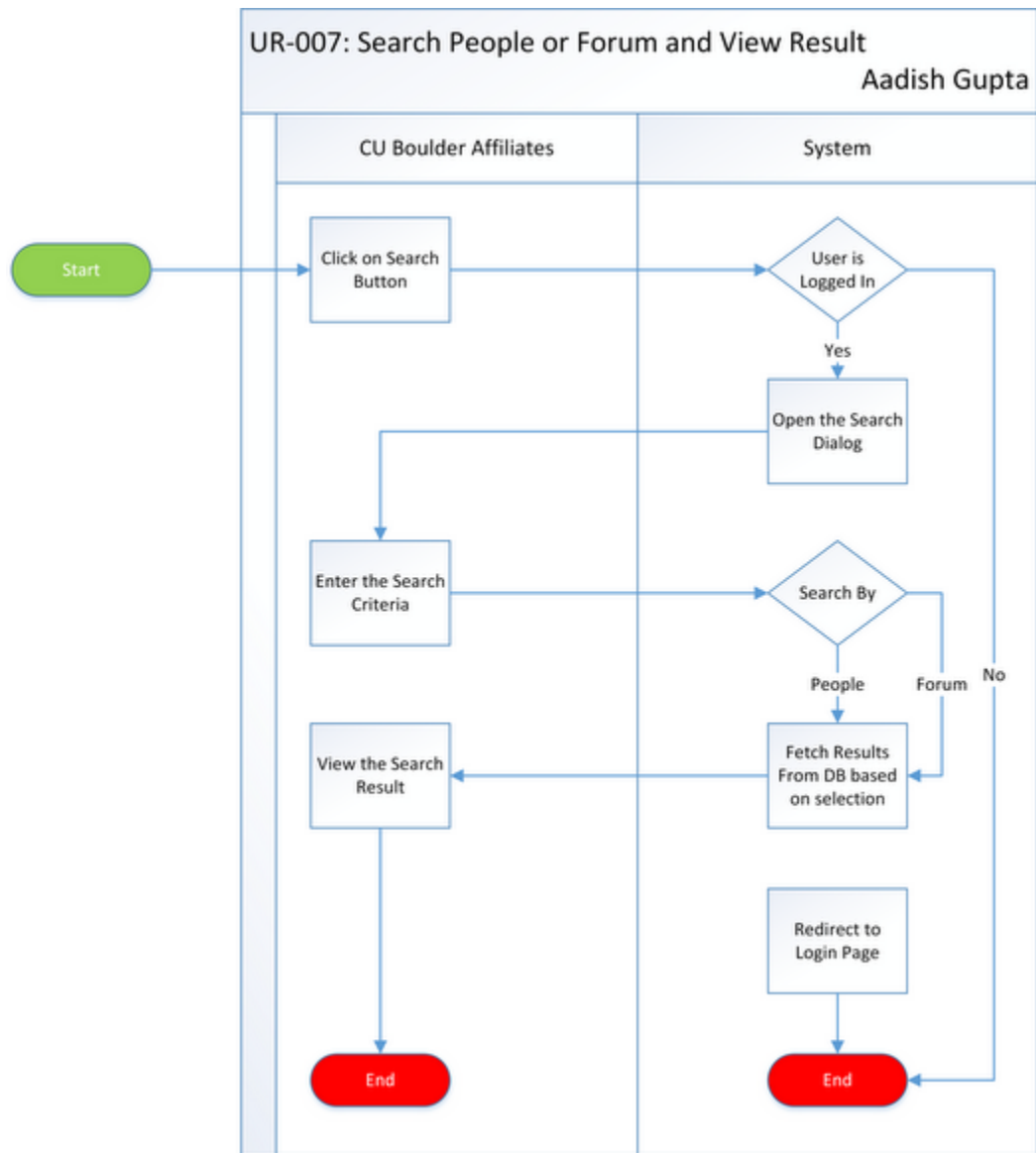
Actors:	CU Boulder Affiliate		
Pre-conditions:	User must be logged in and subscribed to the forum to view the posts belonging to the forum.		
Post-conditions:	Reply to the forum post will be visible to all other users who are subscribed to the forum		
Frequency of Use:	Frequently		
Flow of Events:		Actor Action	System Response
	1	Click on the post link.	A post page where the post and related replies are displayed. A text box will be displayed to the user where user can enter his reply.
	2	Type in the text in the text area and click on Reply button	The reply is saved in the DB and associated with post. This reply will be visible to all other users when they navigate to this post page.
Variations:	User can navigate to the post page either by clicking on a post link at Home page/activity feed or at the forum page.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UR-014
Use Case Name:	User can logout of the application
Description:	User can log-out of the application, when he no longer wants to use the application.

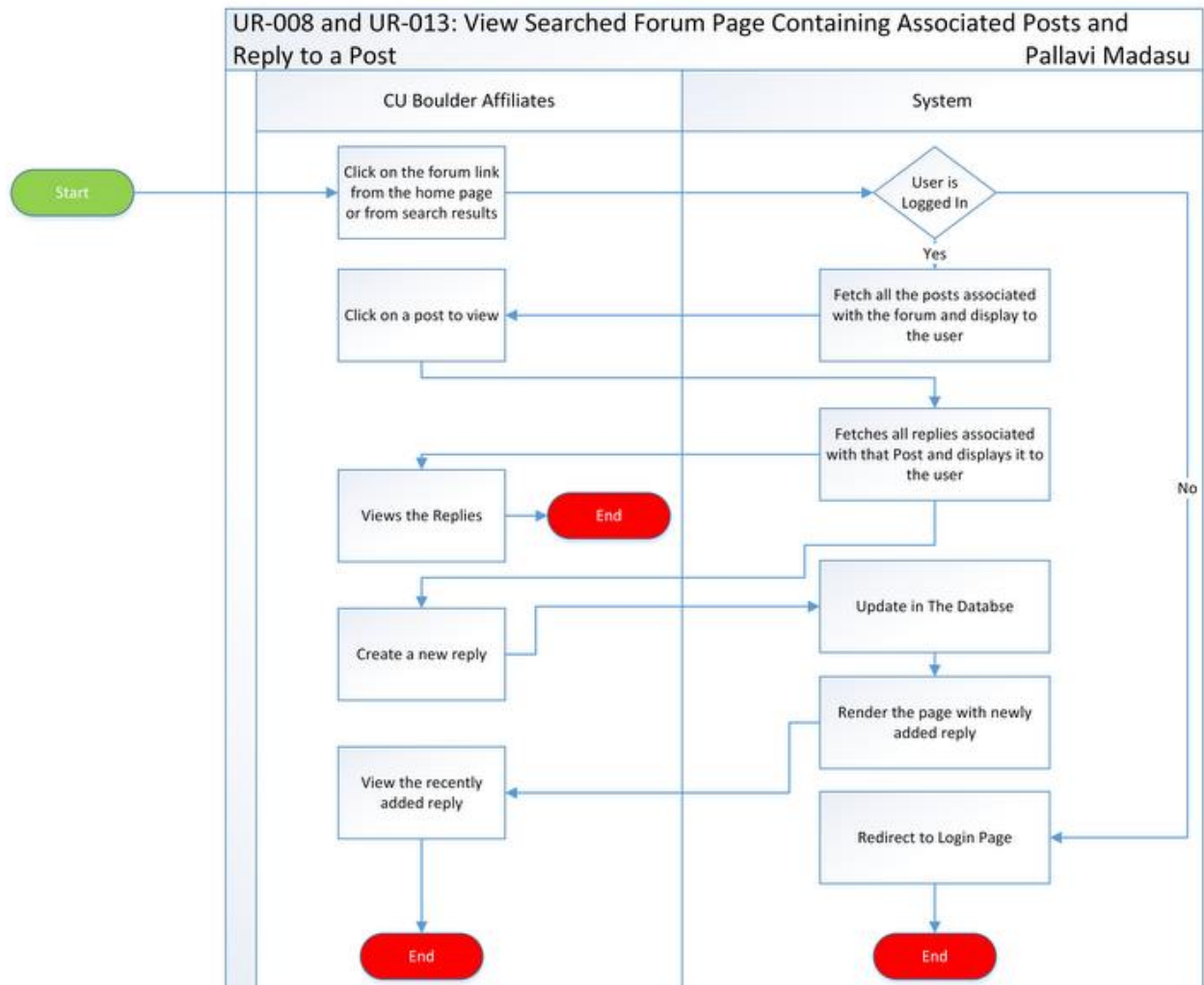
Actors:	CU Boulder Affiliate		
Pre-conditions:	User must be logged in		
Post-conditions:	No pages of the application except sign up page should be accessible after user is logged out.		
Frequency of Use:	Rarely		
Flow of Events:		Actor Action	System Response
	1	Click on the Log Out button.	User's session ends and will be redirected to Log-in page.
Variations:			
Notes and Issues:			
Developer Notes:			

Activity Diagrams

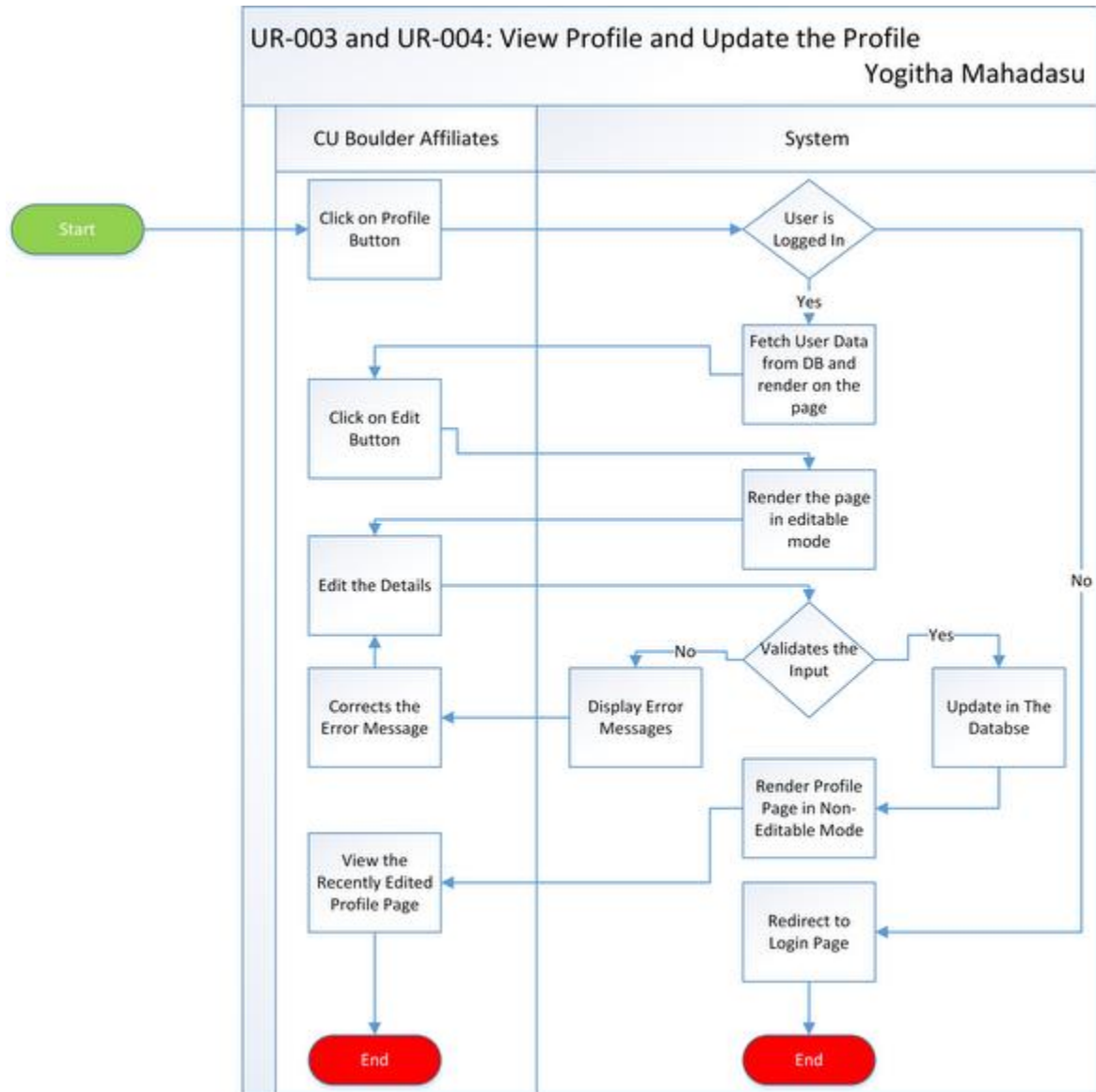
Activity Diagram 1



Activity Diagram 2



Activity Diagram 3

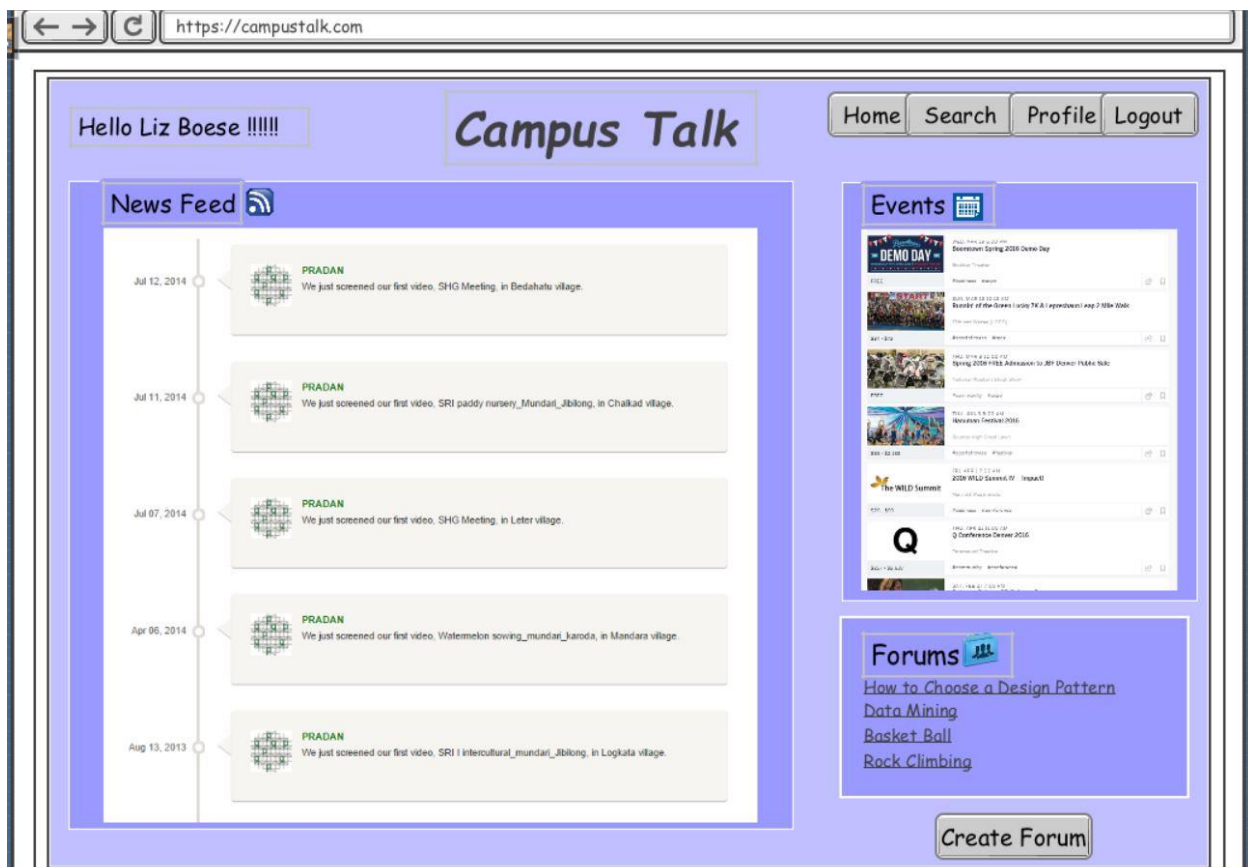


Data Storage

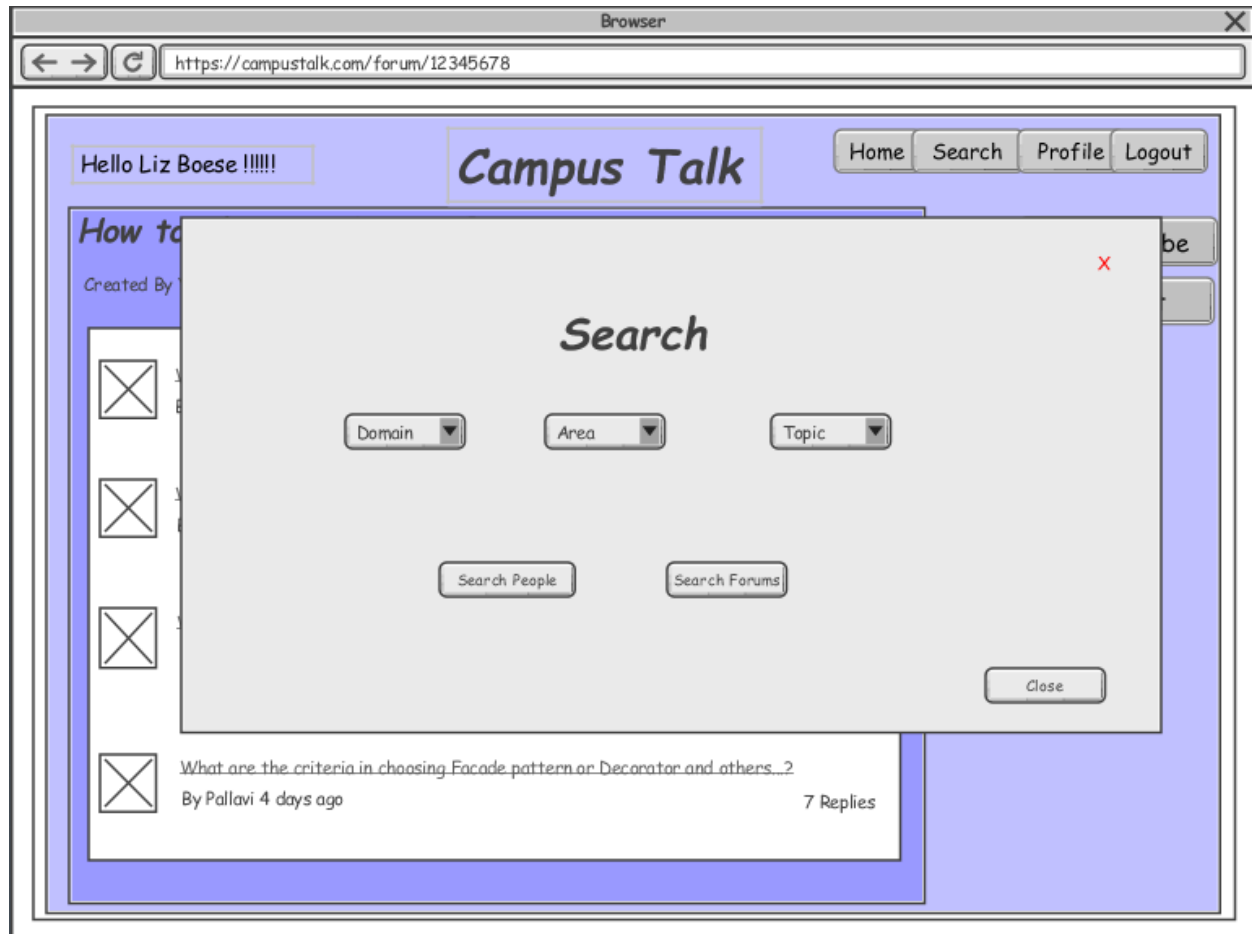
- Relational Database – MySQL
 - The data will be stored as records in the tables of the MySQL Database, which resides on our server deployment.
- **Classes:** **DbHelper** class contains all the methods to interact with the underlying data storage. We modelled this class in a way that it would be easy to change the underlying database with minimal changes in the code. The only code changes required to change the underlying database would be the function in DbHelper (getConnectionToDb) which connects to the database.

UI Mockups

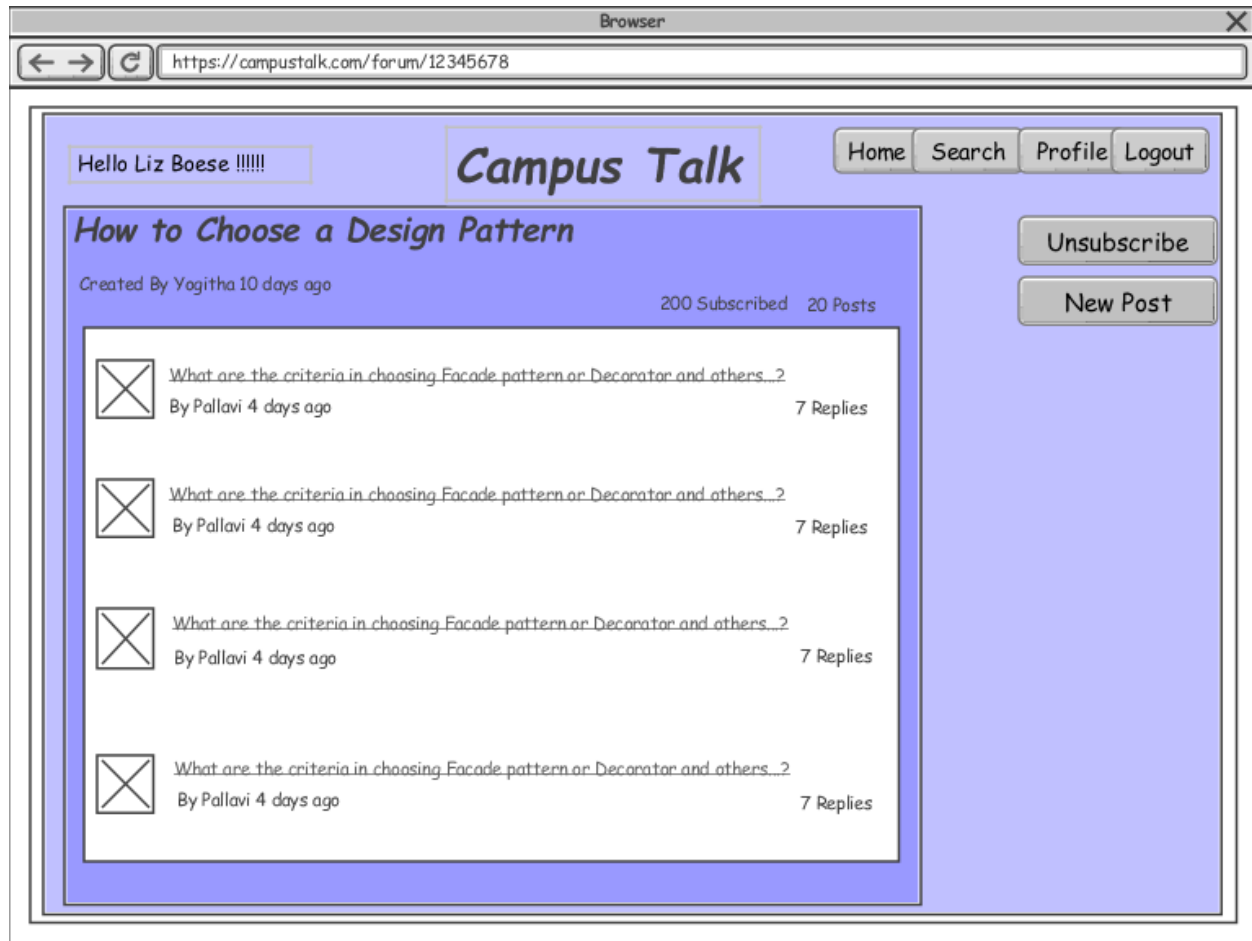
Home Page Screen



Search Screen



Forum Screen

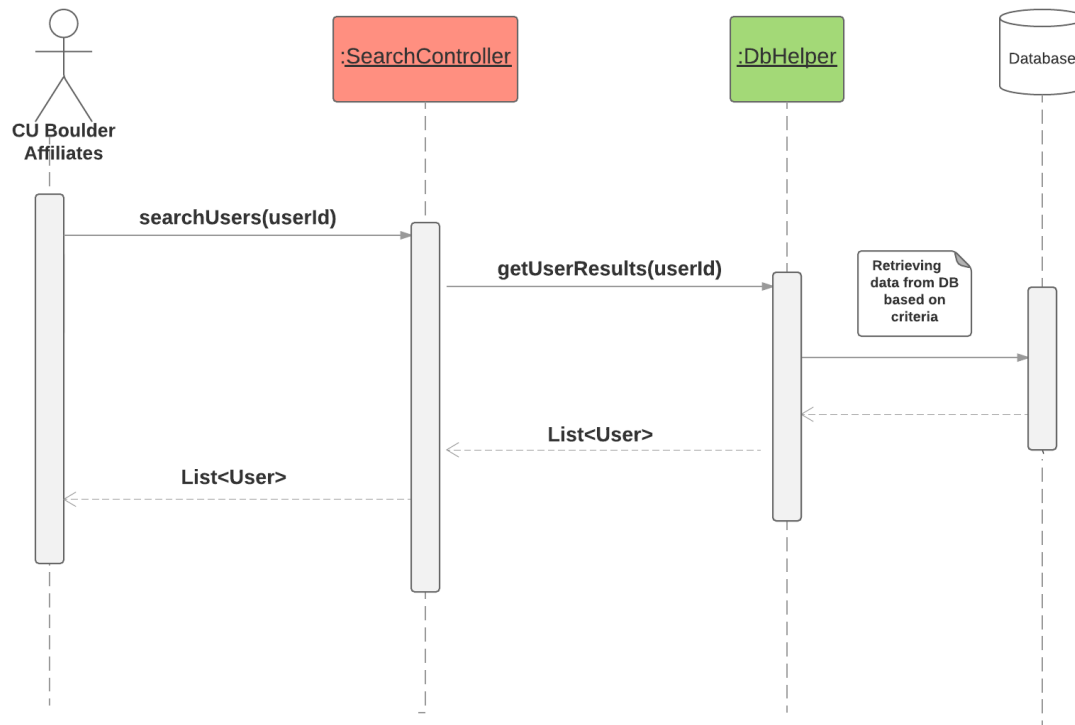


User Interactions

User Interaction 1

UR - 007 : SEARCH PEOPLE

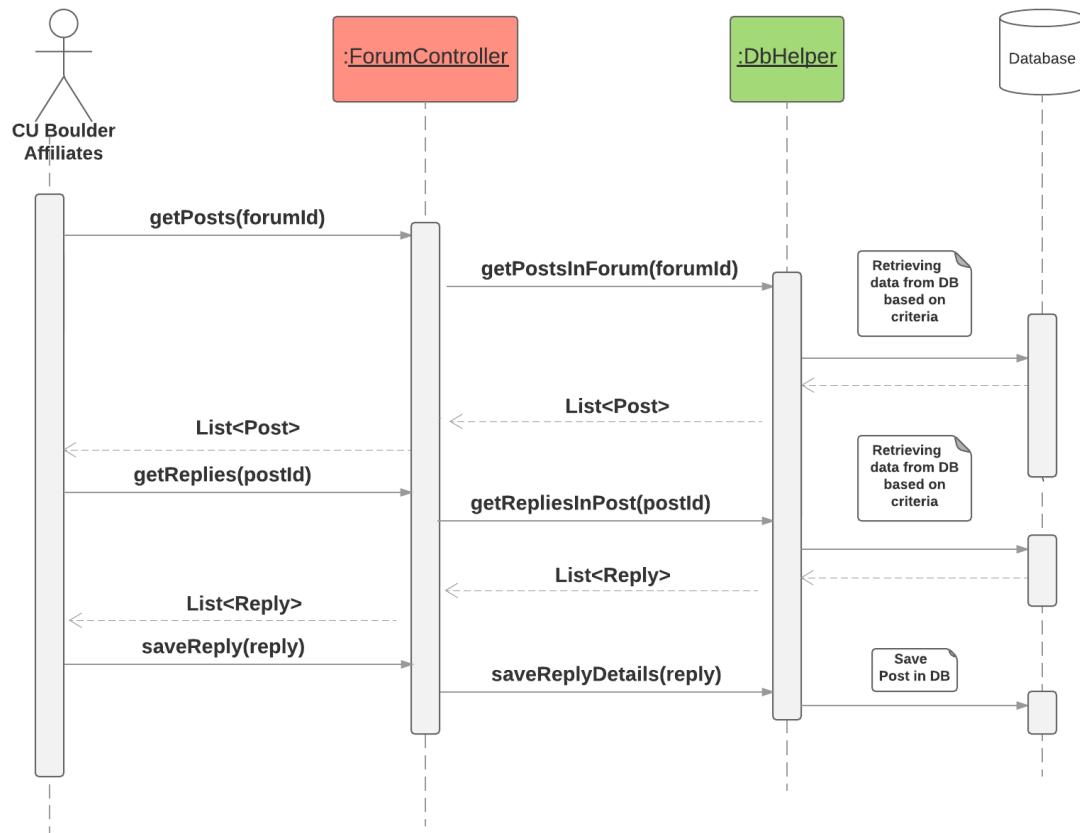
AADISH GUPTA



User Interaction 2

UR - 008, UR - 013 : VIEW POSTS IN FORUM AND REPLY TO A POST

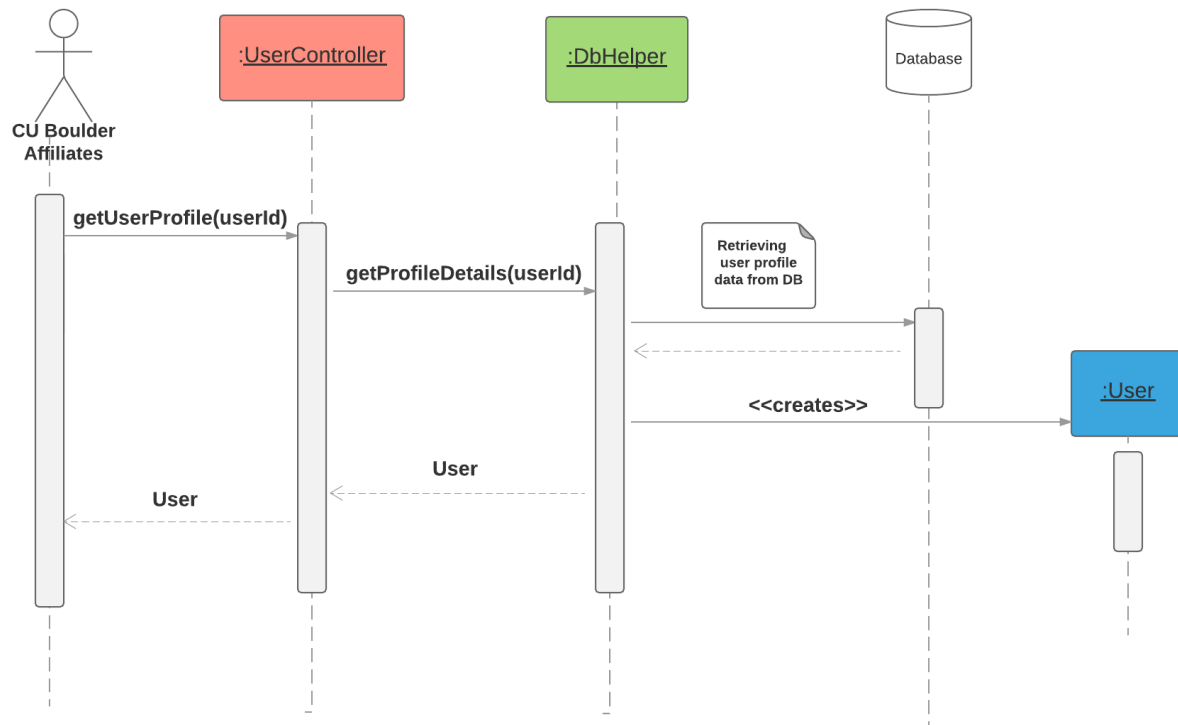
PALLAVI MADASU



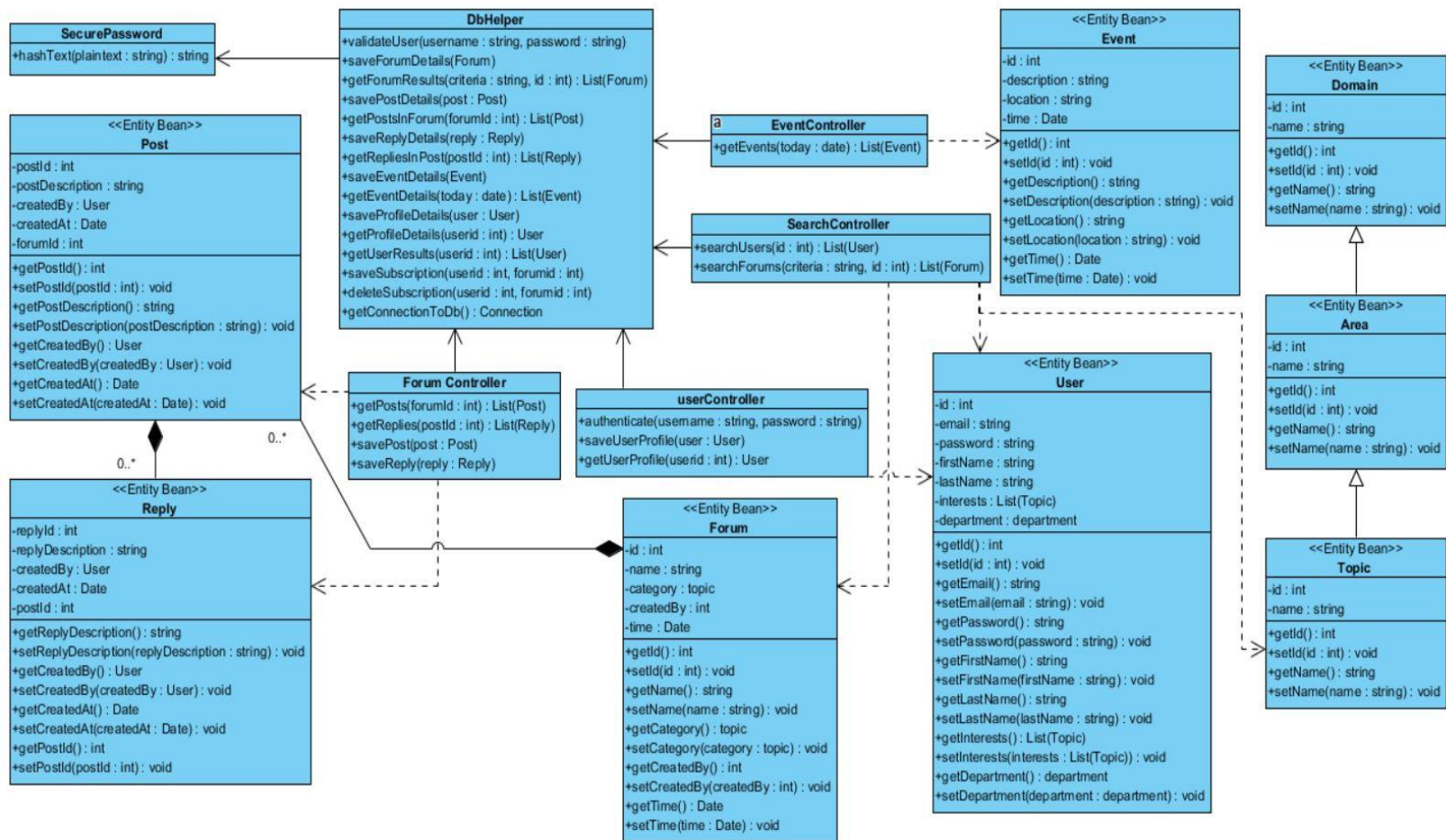
User Interaction 3

UR - 003 : VIEW PROFILE

YOGITHA MAHADASU



Class Diagram



Design Patterns Used:

- **Strategy:** To hash the password, we are using an algorithm which is implemented in a separate class 'SecurePassword'. This algorithm can be changed without modifying the client code using Strategy design pattern.
- **Proxy:** To retrieve/store data from/to Database, client will interact with controllers, which act as wrapper classes for the DbHelper class, which interacts with the underlying database directly.
- **MVC:** We are planning on using Model-View-Controller architectural pattern to separate concerns of the application into business logic, entity data, views(UI) . Using this pattern enables the application to have loose coupling between the UI and backend so that we could extend our application to create native mobile applications.
- **Observer:** There are one-many dependencies between forum and posts, post and replies in a forum. Whenever a new reply pops in, the corresponding post is notified and refreshed with the updated replies. In the same way, whenever a new post is added to the forum, the forum would be updated with the new posts.