

MARK FULLER  
KEVIN KULDA  
CONNOR WOODAHL

PROFESSOR:  
DR. CERNY

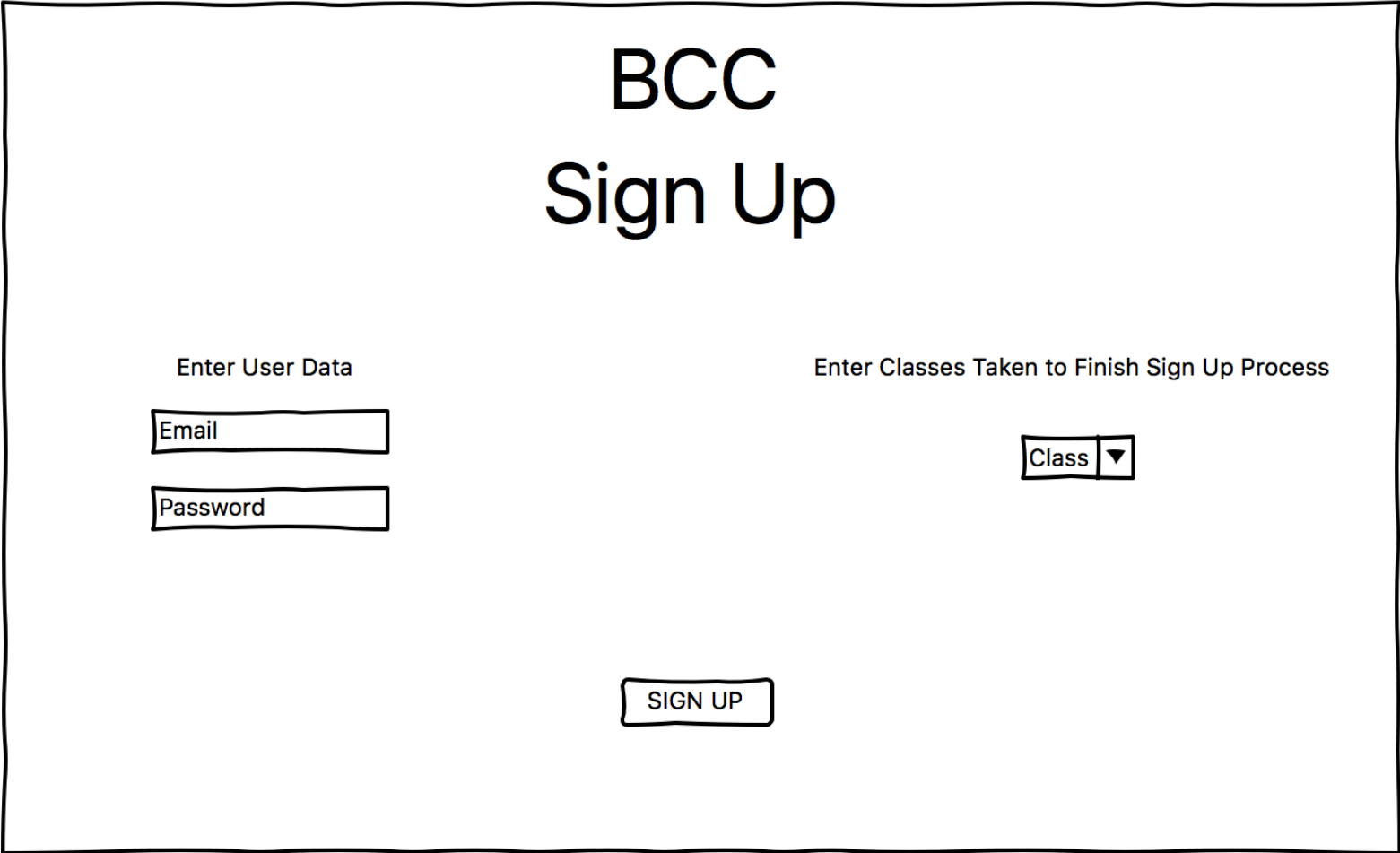


**BCC**

# BAYLOR CLASS CONNECT

- BCC is a tool used by students to rate their current teachers, view reviews of future teachers and contact previous students who have had those teachers

# WIREFRAMES



A hand-drawn wireframe of a sign-up form titled "BCC Sign Up". The form is enclosed in a rectangular border. At the top center, the text "BCC" is above "Sign Up". Below this, the form is divided into two sections. The left section, titled "Enter User Data", contains two input fields: "Email" and "Password". The right section, titled "Enter Classes Taken to Finish Sign Up Process", contains a dropdown menu labeled "Class" with a downward arrow. At the bottom center of the form is a "SIGN UP" button.

BCC  
Sign Up

Enter User Data

Email

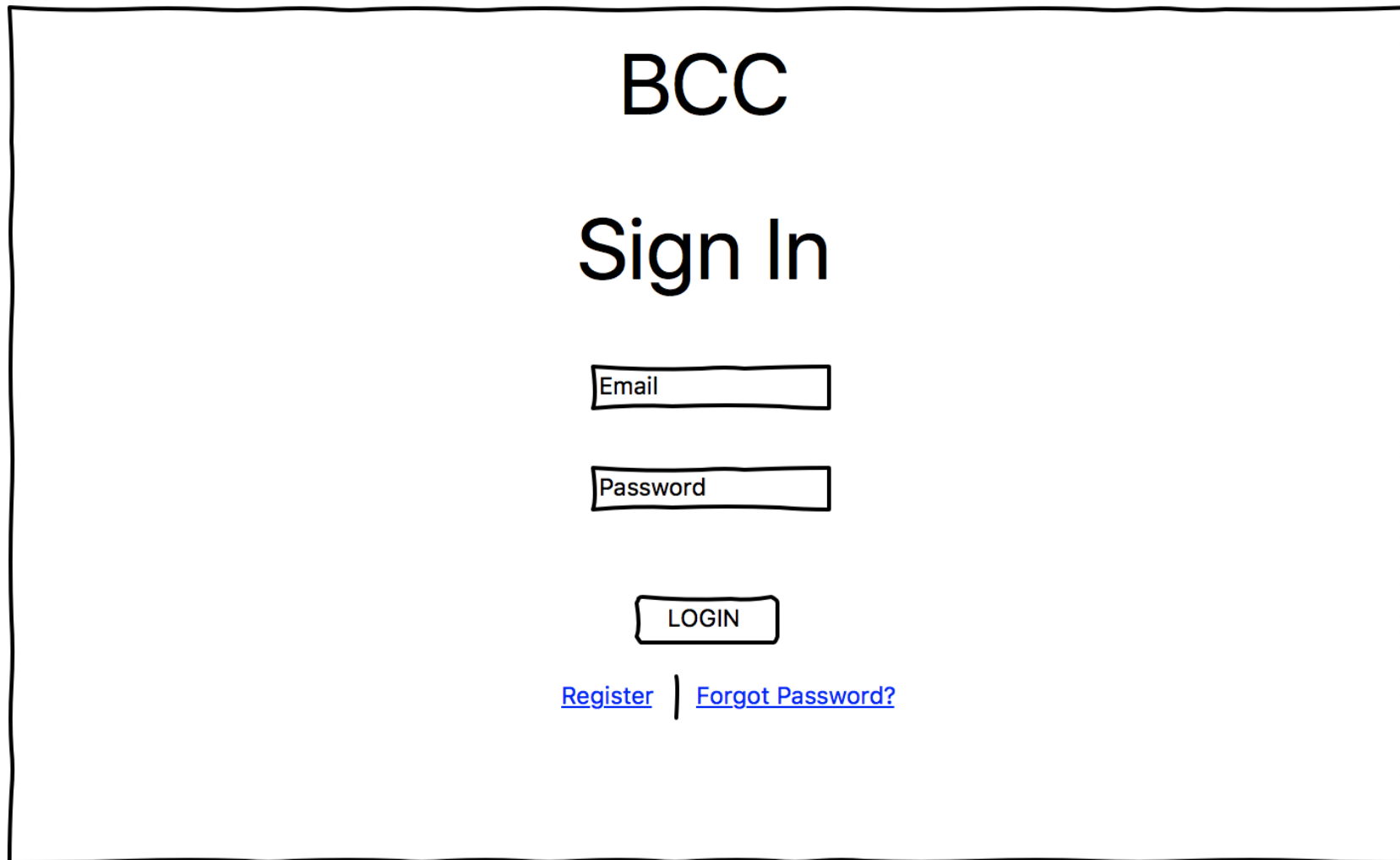
Password

Enter Classes Taken to Finish Sign Up Process

Class ▼

SIGN UP

# WIREFRAMES



A hand-drawn wireframe of a web page for a BCC Sign In interface. The page is enclosed in a rectangular border. At the top center is the text "BCC". Below it is the title "Sign In". Under the title are two input fields: the first is labeled "Email" and the second is labeled "Password". Below these fields is a button labeled "LOGIN". At the bottom of the form area are two links: "[Register](#)" and "[Forgot Password?](#)", separated by a vertical line.

BCC

Sign In

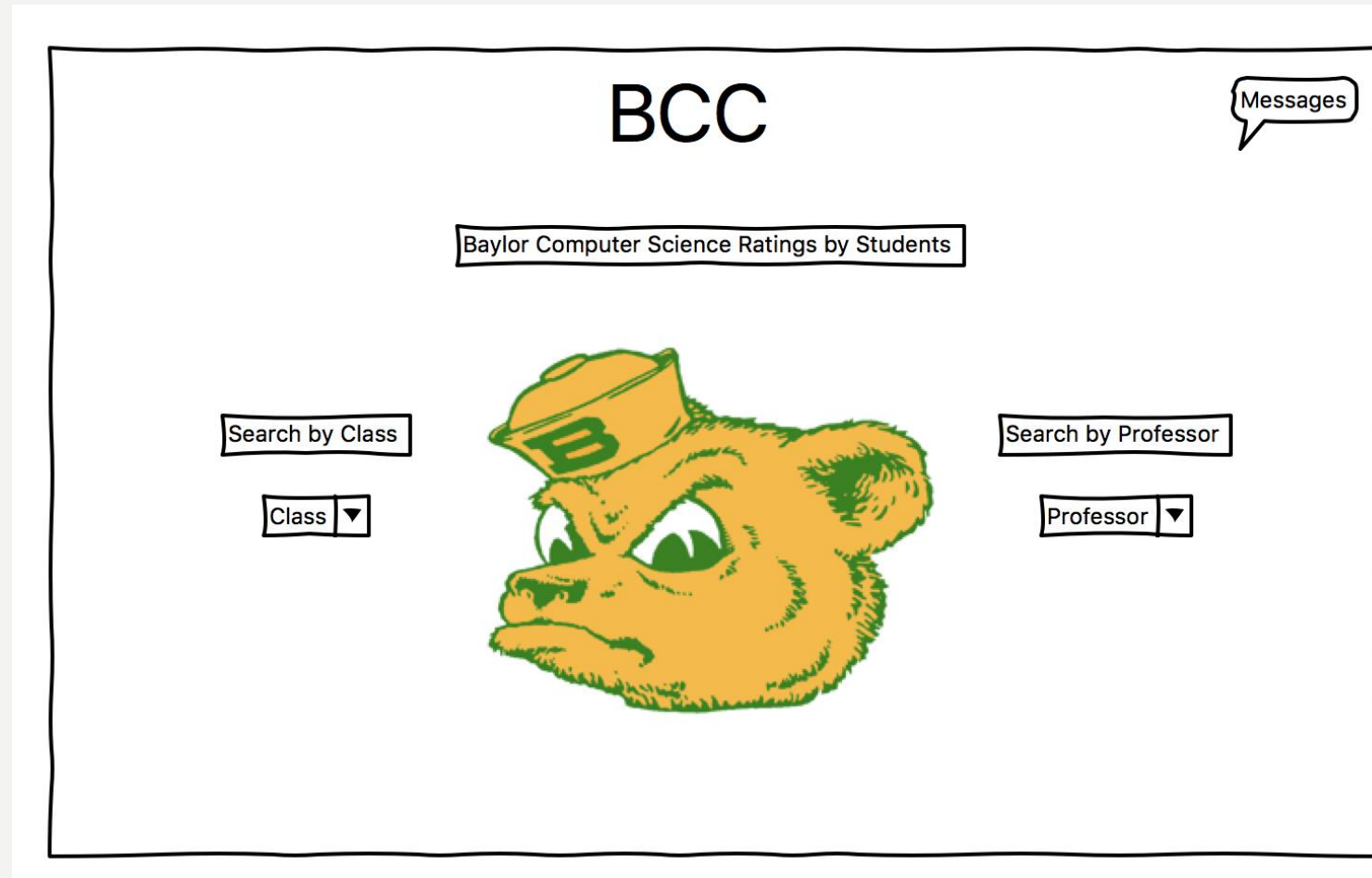
Email

Password

LOGIN

[Register](#) | [Forgot Password?](#)

# WIREFRAMES



# WIREFRAMES

BACK

(Teacher)

Email: example@gmail.com

Classes Taught	Ratings	Number of Reviews
<a href="#">Class 1</a>	80	15
<a href="#">Class 2</a>	40	5
<a href="#">Class 3</a>	95	3

# WIREFRAMES

(Class)		
<a href="#">BACK</a>		
Teachers	Ratings	Number of Reviews
<a href="#">Teacher 1</a>	80	15
<a href="#">Teacher 2</a>	40	5
<a href="#">Teacher 3</a>	95	3


# WIREFRAMES

BACK

(Teacher) (Class)

Email: example@gmail.com

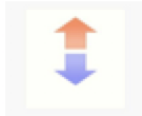
Score	Helpfulness	Teaching Ability	Workload
83	50	99	45

4

UserID 

Message

Example Review .....

1

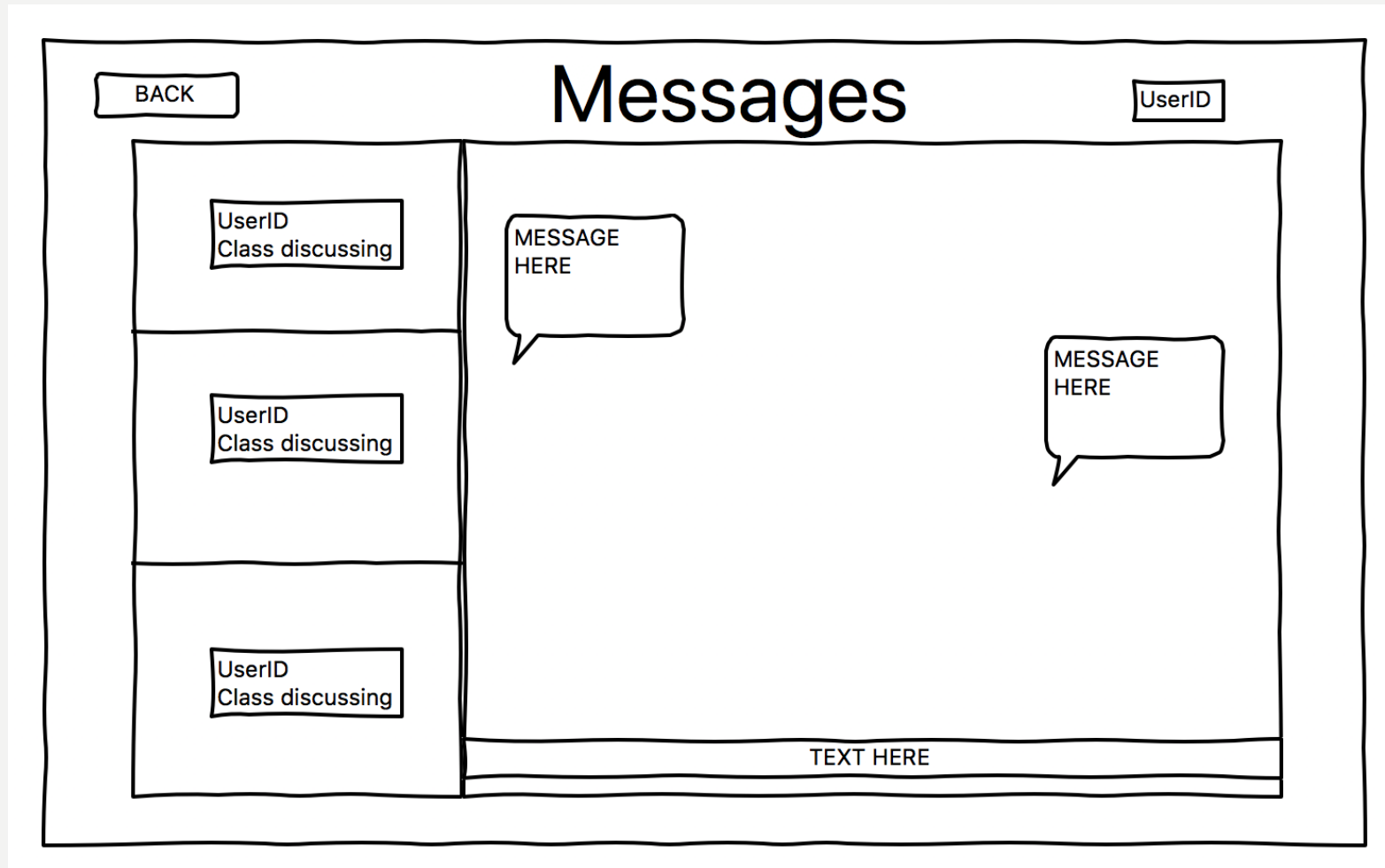
UserID 

Message

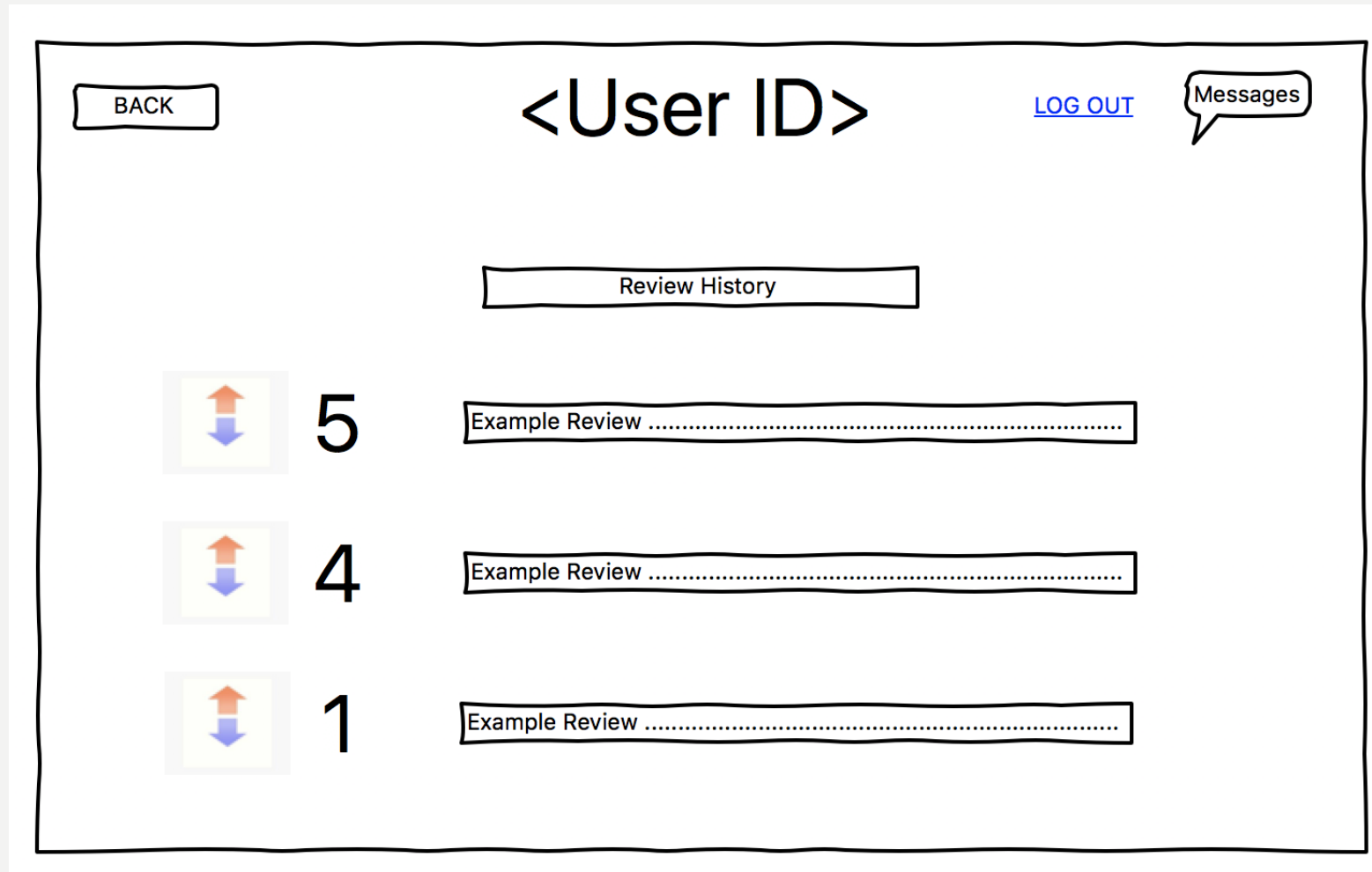
Example Review .....



# WIREFRAMES



# WIREFRAMES



# REQUIREMENTS

## Functional Requirements

- Users need to be able to register an account.
- Users must be able to search based on class.
- Users must be able to see professor options after searching the class.
- Users must be able to search based on professor.
- Users must be able to write a review on a professor.
- Users must be able to upvote or downvote a review.
- Users must be able to send a message to the author of the review.
- The reviewer must be able to respond to a message.
- User must be able to log out.
- The application must be able to sanitize user input.
- User must be able to navigate bidirectionally in this application.
- Users need to be able to delete an account.
- Users must be able edit a review.

## Non-Functional Requirements

- Messages must be able to be sent anonymously
- User data must be kept private
- Application must be fast for the average user
- Application must handle traffic of at most 20 users at once
- User interface should be intuitive
- User interface should be graphically pleasing

# ACTORS

User



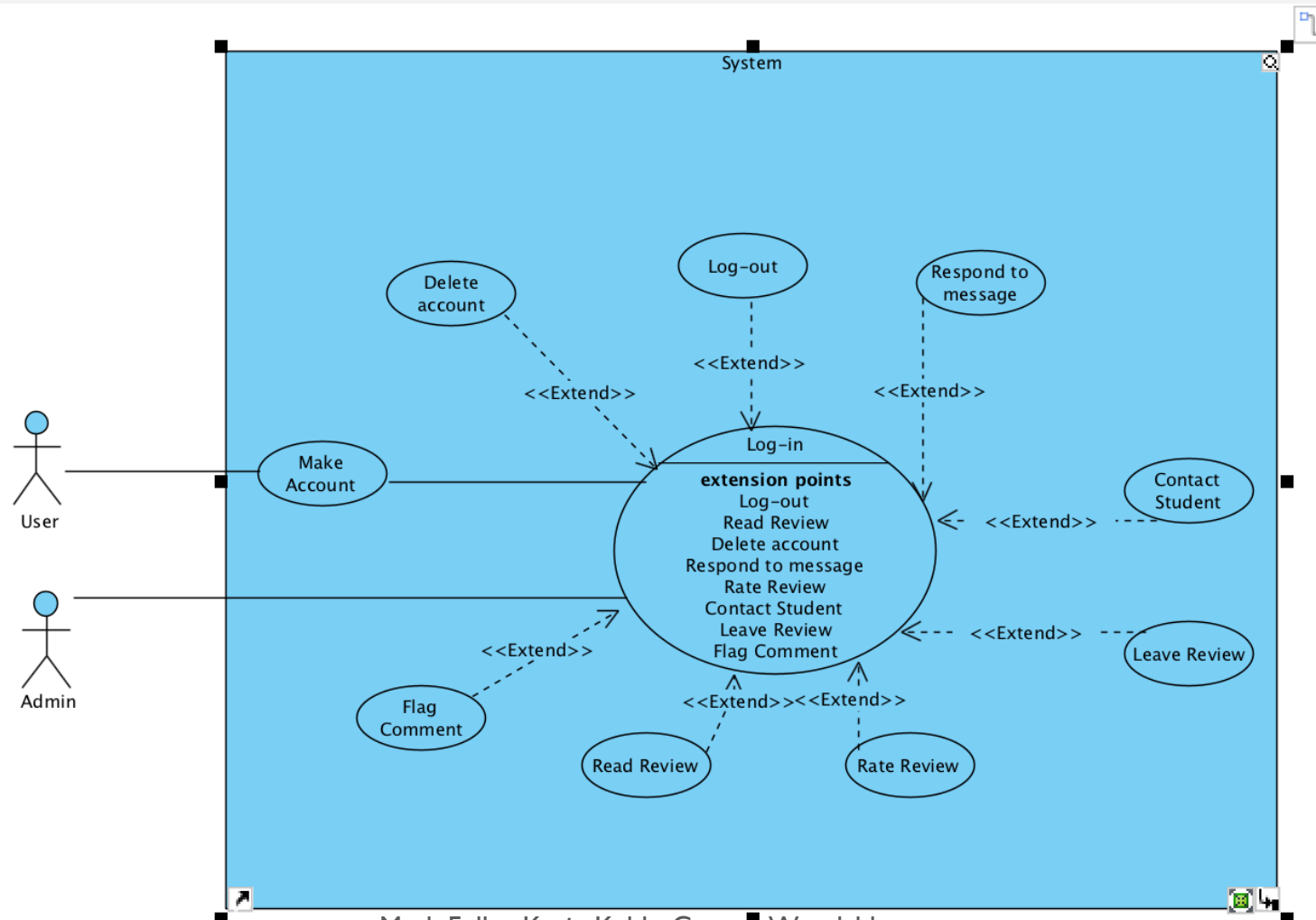
Admin



System



# USE CASE DIAGRAM



# USE CASE DIVISION

- Use Cases handled by Kevin– Make Account, Respond to a Message, Flag Comment
- Use Cases handled by Mark – Leave Review, Rate Review, Read Review
- Use Cases handled by Connor – Contact Student, Edit Review, Delete Account, Log Out

# MAKE AN ACCOUNT

Make an Account

Scope: GCC Application

Level: User Goal

Primary Actor: Student

Stakeholders and Interests:

- Student User: Wants username and password stored and hashed for security.

Preconditions: Student has loaded the site

Success Guarantee: Students data is hashed and saved into database. Immediate access to the application is granted.

Main Success Scenario:

- Student loads application.

- Student requests a new account.

- Student enters Username, Password and, email, etc.

- System updates valid students list with new student.

- Student Logs in.

Alternative Flows:

- a. At any time the student closes the application.

- I. Interaction ends.

- b. At any time system fails.

- I. Student will restart application.

- c. While student is on register page, they would like to login with valid credentials(the realized they had an account already)

- I. Student clicks `return to login` and logs in with valid username and password.

- d. Username and password from account creation do not work.

- I. Prompt student to reset password with security question.

- I.1. Student answers security question correctly

- I.1.1. Student resets password.

- I.2. Student does not answer security question correctly

- I.2.1 Student is returned to login page.

Mark Fuller, Kevin Kulda, Connor Woodahl ---  
<https://github.com/markfuller/BNU>

# LEAVE REVIEW

Leave Review

Scope: GCC Application

Level: User Goal

Primary Actor: Student

Stakeholders and Interests:

Student User:

Wants Username and Password to be recognized.

Wants review to be saved for others to view

Student has valid login credentials

Preconditions:

Student has loaded the program.

Success Guarantee:

Student review is made available to other users on the site.

Main Success Scenario:

Student loads application.

Student logs in.

Student selects professor they would like to review from drop-down menu

Student is brought to professors review page

Student writes review

Student saves review

Alternative Flows:

a. At any time the student closes the application.

1. Interaction ends.

b. At any time system fails.

1. Student will restart application.

c. Student picks wrong professor

1. Student clicks 'back'

2. Selects correct professor from drop-down list

d. Username and password do not work.

1. Prompt student to reset password with security question.

1.1. Student answers security question correctly

1.1.1. Student resets password.

1.2. Student does not answer security question correctly

1.2.1 Student is returned to login page.



# CONTACT STUDENT

Contact Student

Scope: GCC Application

Level: User Goal

Primary Actor: Student

Stakeholders and Interests:

Student User:

Wants Username and Password to be recognized.

Wants review to be saved for others to view

Wants contact with other Student to be over message board

Preconditions:

Student has loaded the program.

Student has valid login credentials

Success Guarantee:

Student is able to message other Student user (two-way anonymity)

Main Success Scenario:

Student loads application.

Student logs in.

Student selects professor they would like to view reviews of

Student is brought to professors review page

Student selects, contact reviewer button

Student creates message

Student Sends message

Opposing student responds to message

Message Notification shows up on login of user

Alternative Flows:

a. At any time the student closes the application.

1. Interaction ends.

b. At any time system fails.

1. Student will restart application.

c. Student picks wrong professor

1. Student clicks 'back'

2. Selects correct professor from drop-down list

d. Username and password do not work.

1. Prompt student to reset password with security question.

1.1. Student answers security question correctly

1.1.1. Student resets password.

1.2. Student does not answer security question correctly

1.2.1 Student is returned to login page.

# CASUAL USE CASES

## **Respond to a Message**

### ***Main Success Scenario***

The user clicks on the message board icon on the main page. Selecting the message they want to respond to, they use the messaging capability of the application to respond to the message.

### ***Alternative Success Scenario***

## **Edit Review**

### ***Main Success Scenario***

The user logs in and navigates to their profile where all their reviews are listed. Using the editor, they change the content of their review and save changes.

*Alternative Success Scenario* The user logs in and navigates to their review by selecting the professor and finding their review under the professors page. Using the editor, they change the content of their review and save changes.

## **Delete Account**

### ***Main Success Scenario***

While the user is logged in, the user navigates to their user profile settings and select the delete account button. After confirming your selection they are logged out.

### ***Alternative Success Scenario***

## **Rate Review**

### ***Main Success Scenario***

User logs in with valid credentials. Selects the professor and class to read reviews for. Clicking the up-vote button if the review is accurate will increase the reviews score.

### ***Alternative Success Scenario***

User logs in with valid credentials. Selects the professor and class to read reviews for. Clicking the down-vote button if the review is inaccurate will decrease the reviews score.

## **Flag Comment**

### ***Main Success Scenario***

User logs in with valid credentials. Selects the professor and class to read reviews for. If a review is inappropriate or uses vulgar language, the users flags it.

### ***Alternative Success Scenario***

## **Read Review**

### ***Main Success Scenario***

The user will login with valid credentials. Using the navigation page, selects a professor and the desired class to load reviews for. Reviews load in chronological order.

### ***Alternative Success Scenario***

User logs in with valid credentials. Using the navigation page, selects a class and the desired professor to load reviews for. Reviews load in chronological order.

## **Log Out**

### ***Main Success Scenario***

While the user is logged in, the user navigates to their user profile settings and logs out.

### ***Alternative Success Scenario***

The user is automatically logged out when they close the application.

# GIT HUB PAGE

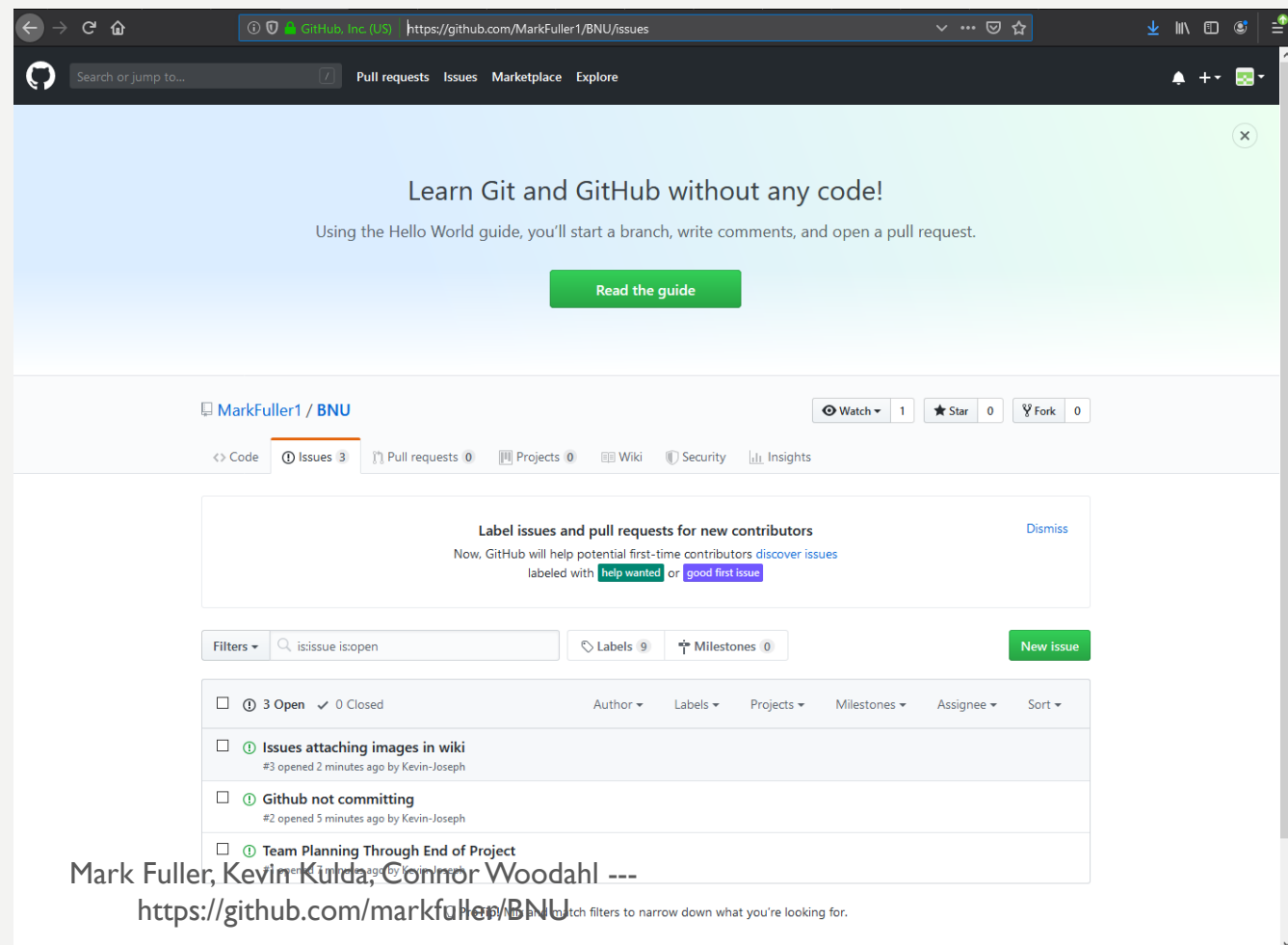
Please see our Github page for the current versions of our development materials and the full repository of our project artifacts.

<https://github.com/MarkFullerI/BNU/>























# ISSUE TRACKING

- Please see our issue tracking repository on Github at the link below:

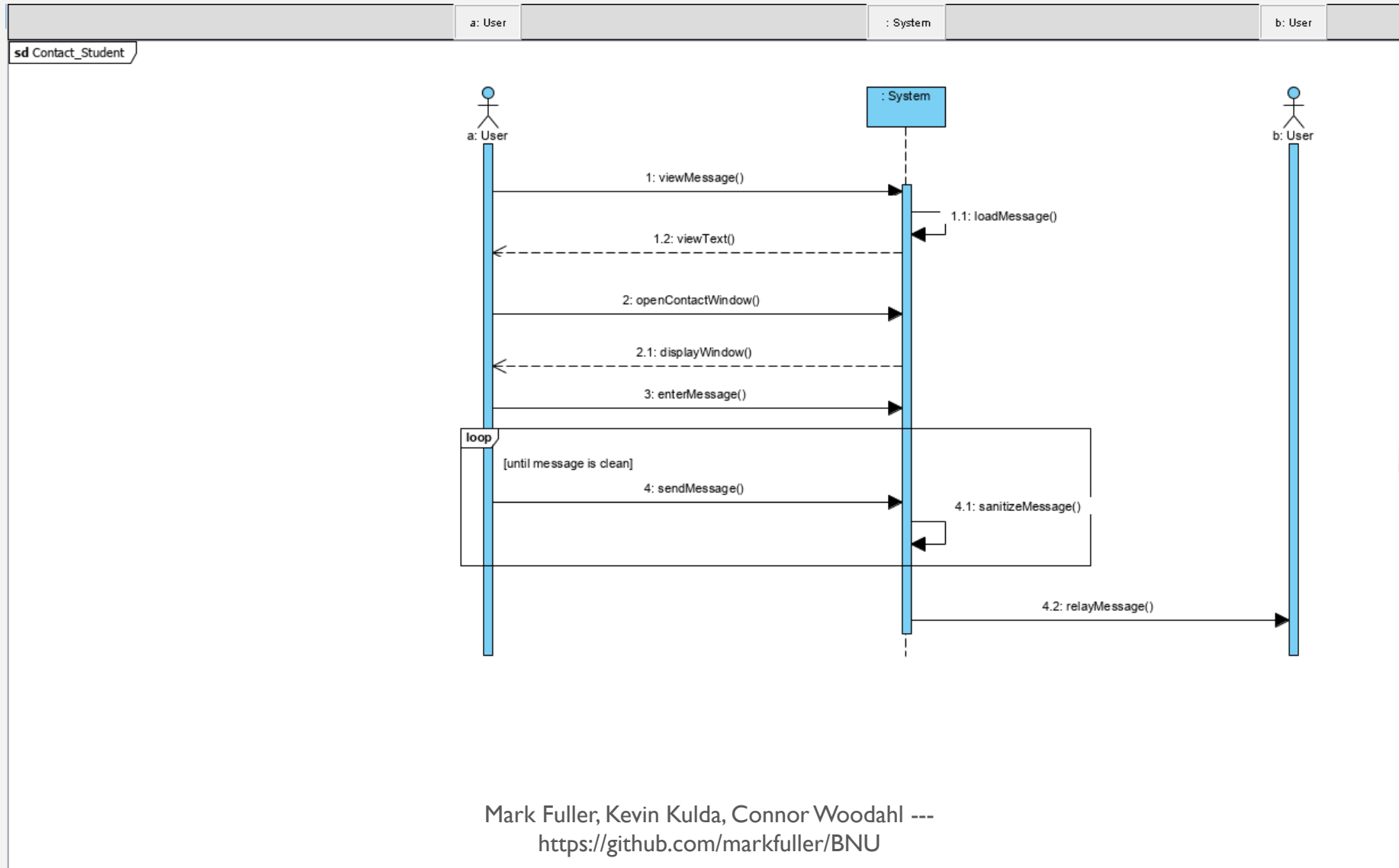
<https://github.com/MarkFuller1/BNU/issues>



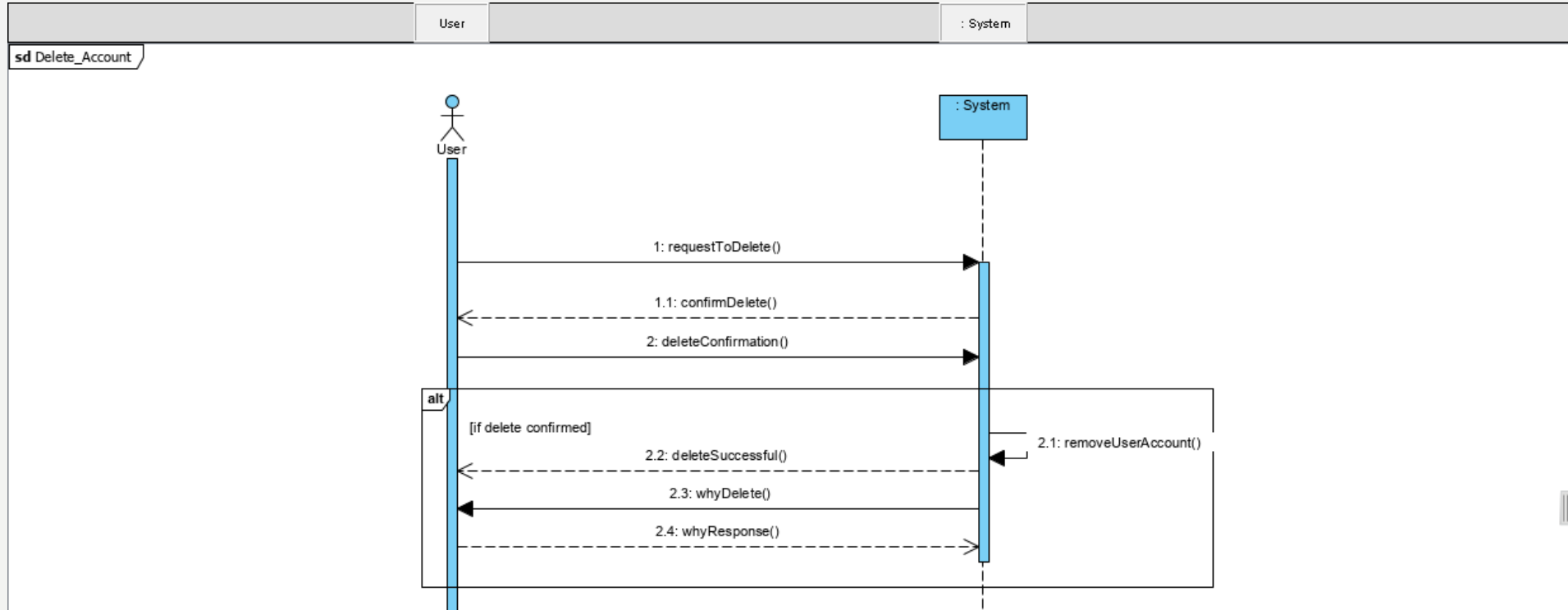
# TRACEABILITY MATRIX

(13) Requirement		Sanitize user input.	Respond to a messag...	Able to log out.	Navigate bidirectionally.	Search based on class.	Search based on prof...	See professor options ...	Message the author of...	Upvote or downvote a ...	View all reviews	Write a review on a pr...	Delete an account.	Register an account.
(9) Use Case														
 Contact Student									✓					
 Delete account													✓	
 Flag Comment		✓												
 Leave Review					✓	✓	✓	✓				✓		
 Log-out				✓										
 Make Account														✓
 Rate Review					✓	✓	✓	✓		✓				
 Read Review					✓	✓	✓	✓			✓			
 Respond to message			✓											

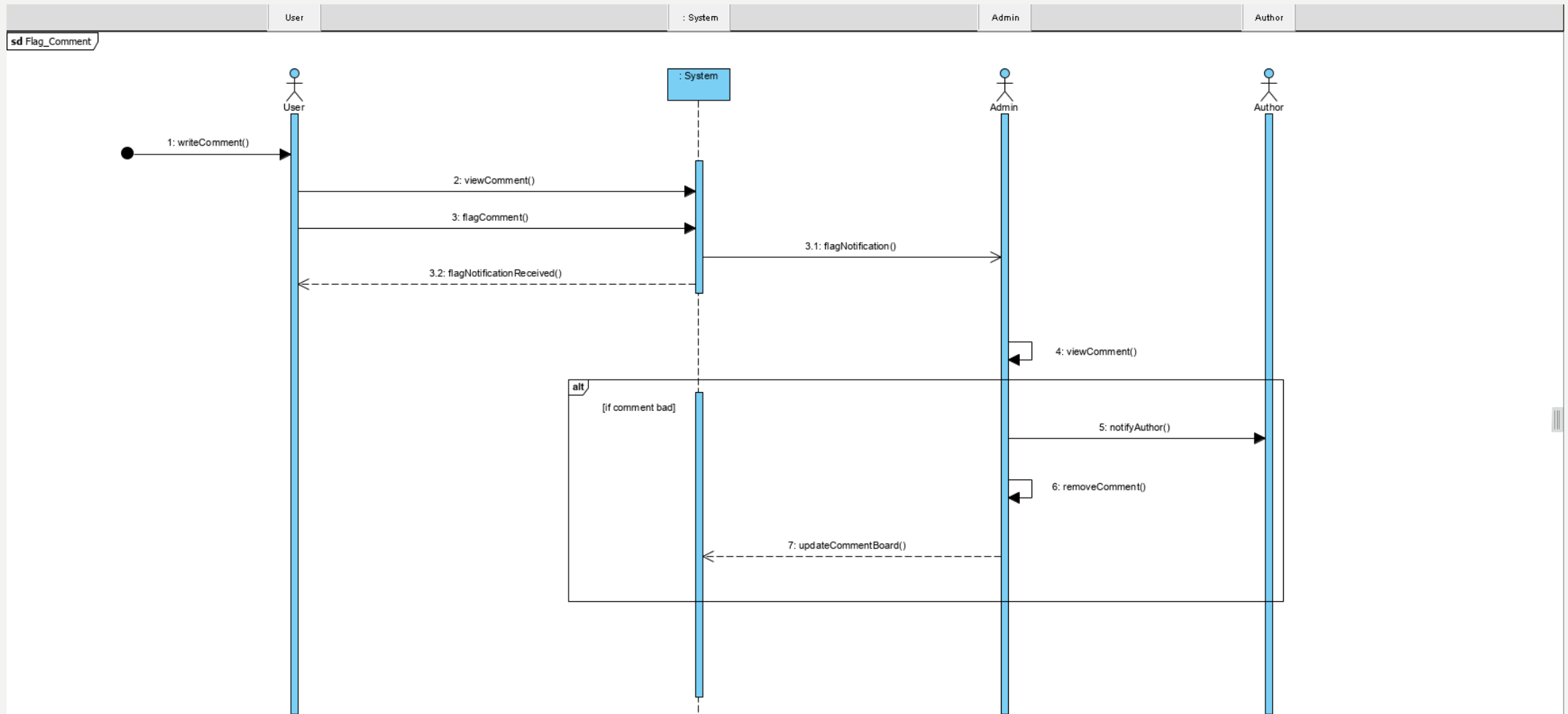
# SSD



# SSD

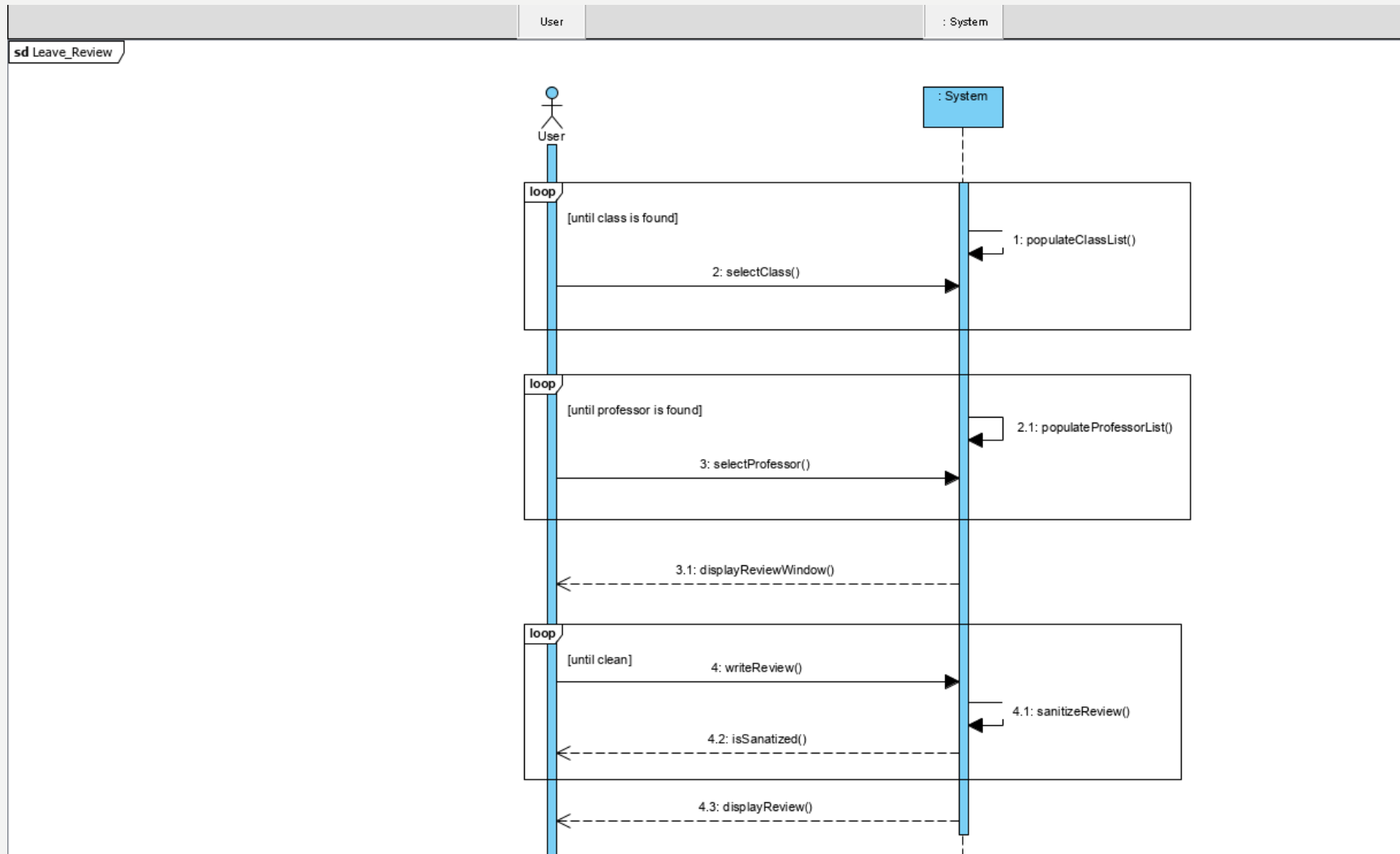


# SSD

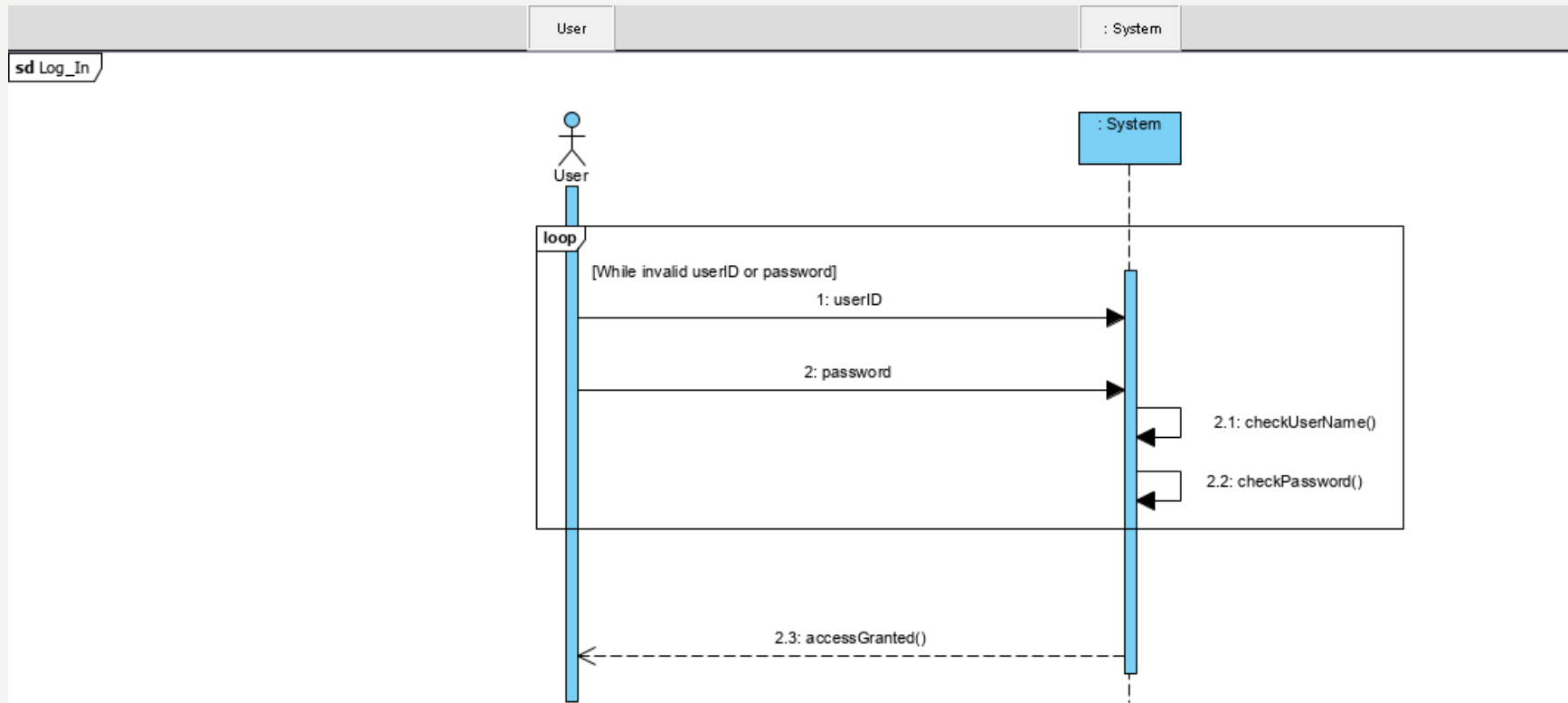




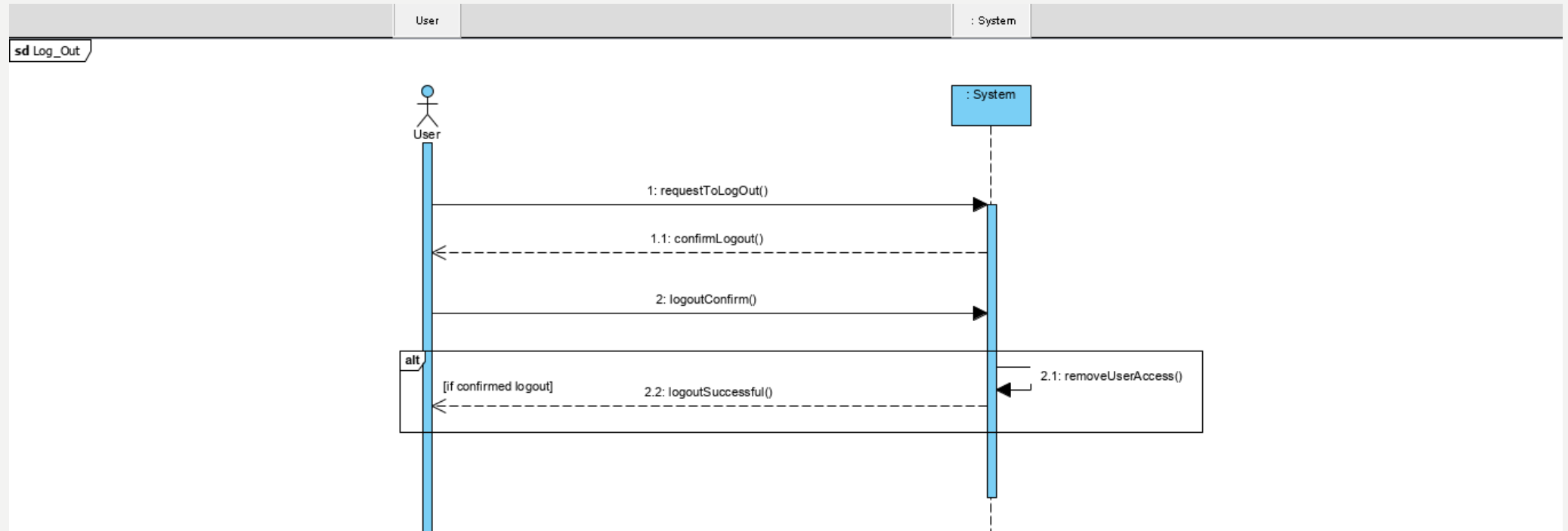
# SSD



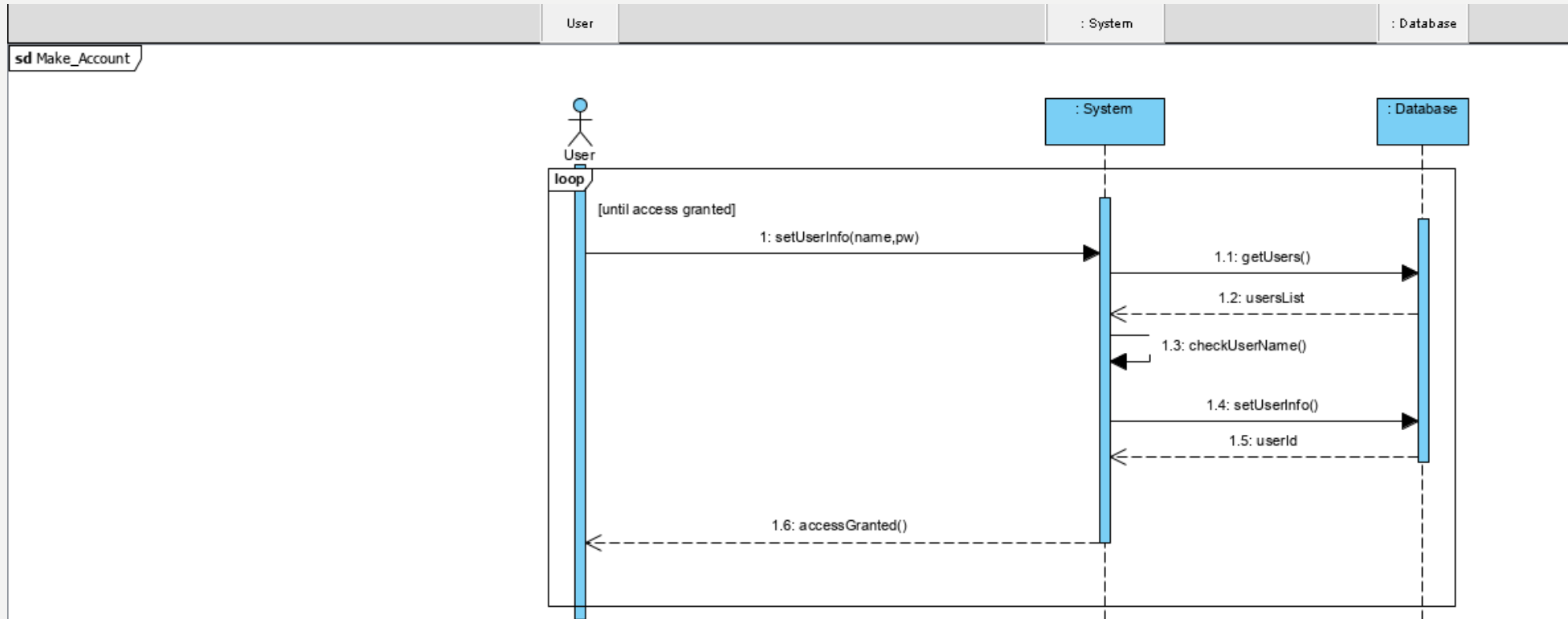
# SSD



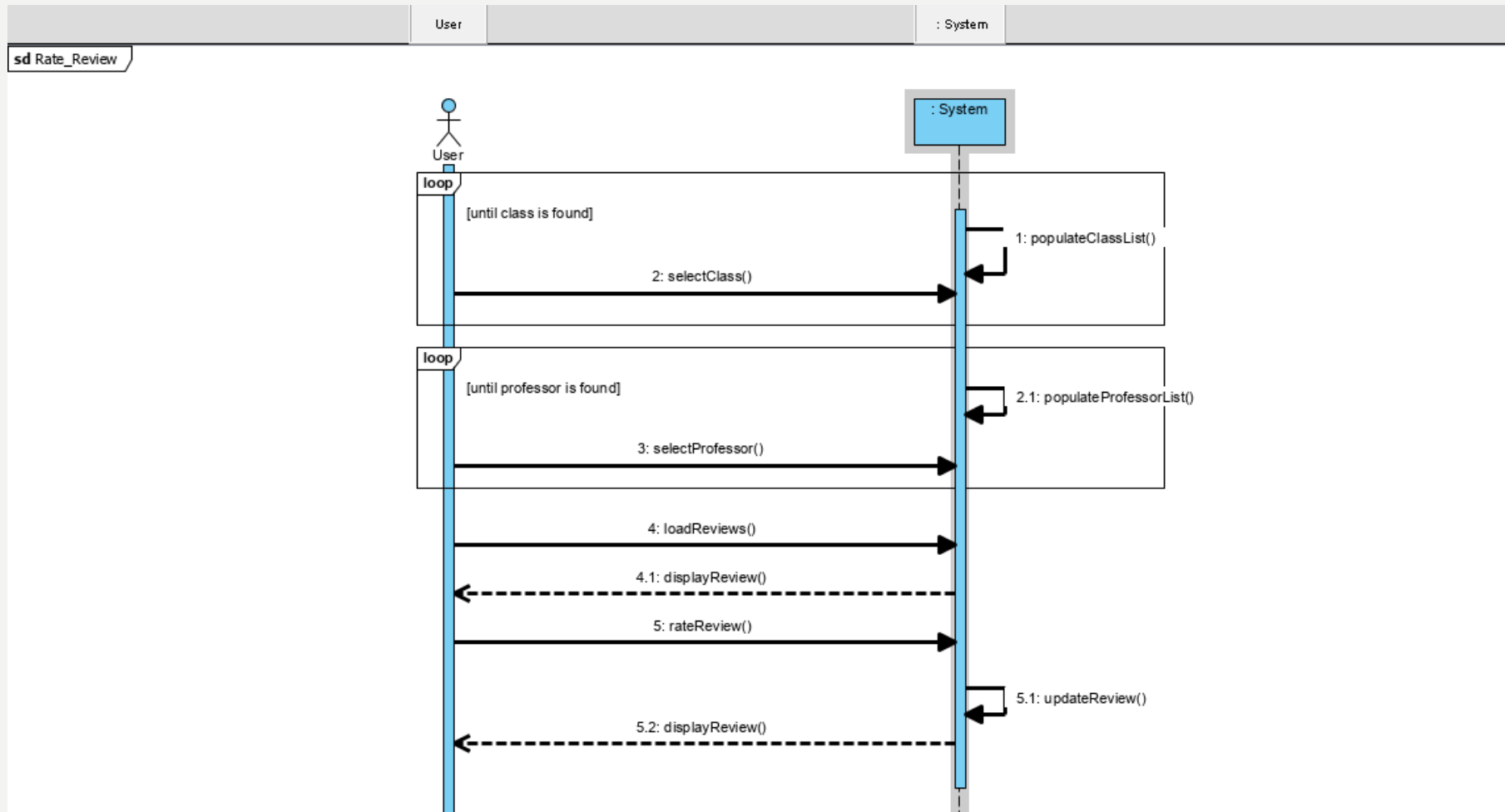
# SSD



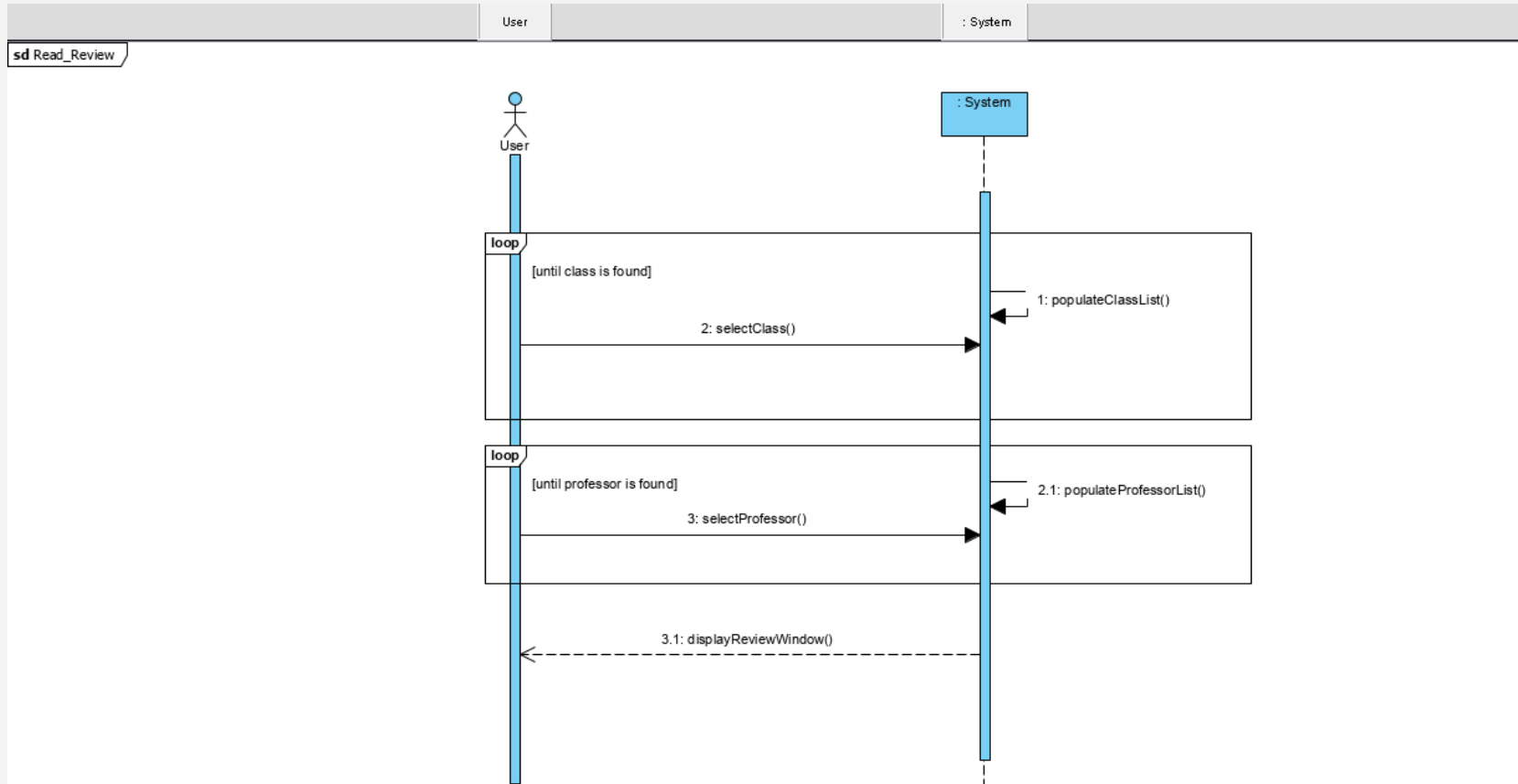
# SSD



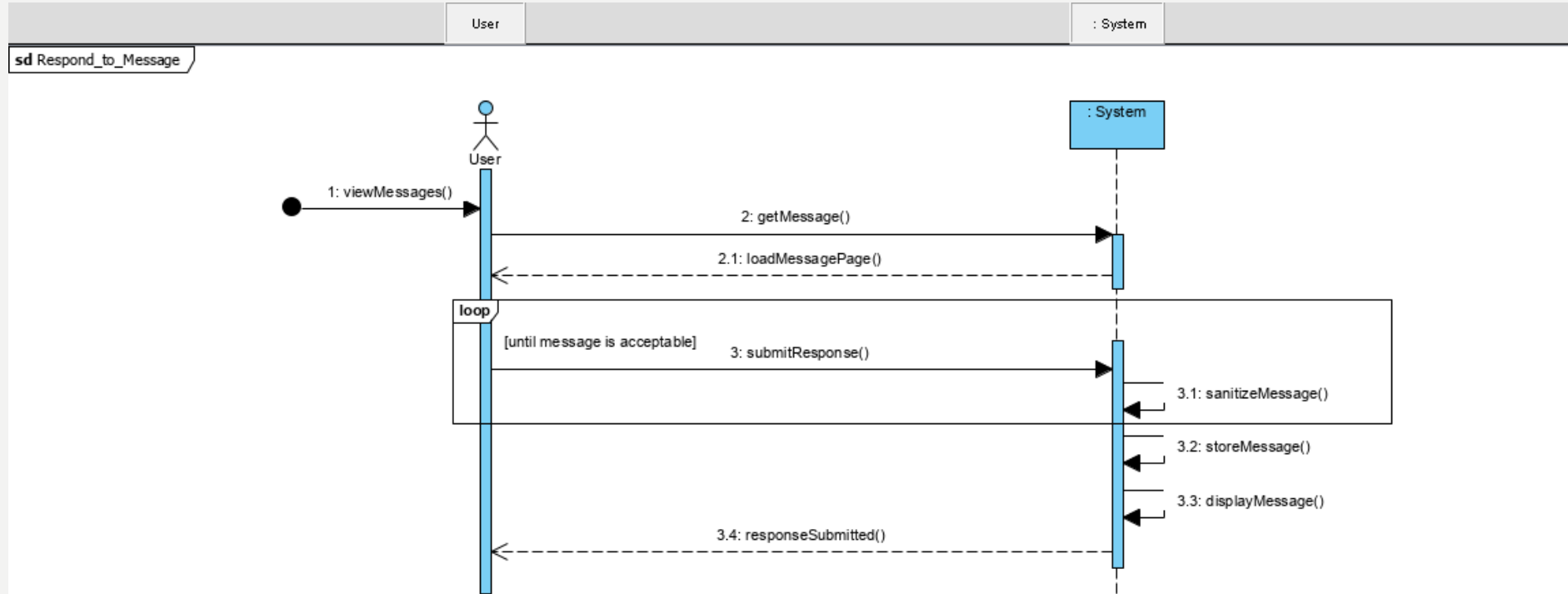
# SSD



# SSD



# SSD



# OPERATIONS

Defining the system operations helps us plan how we will implement the BCC application.

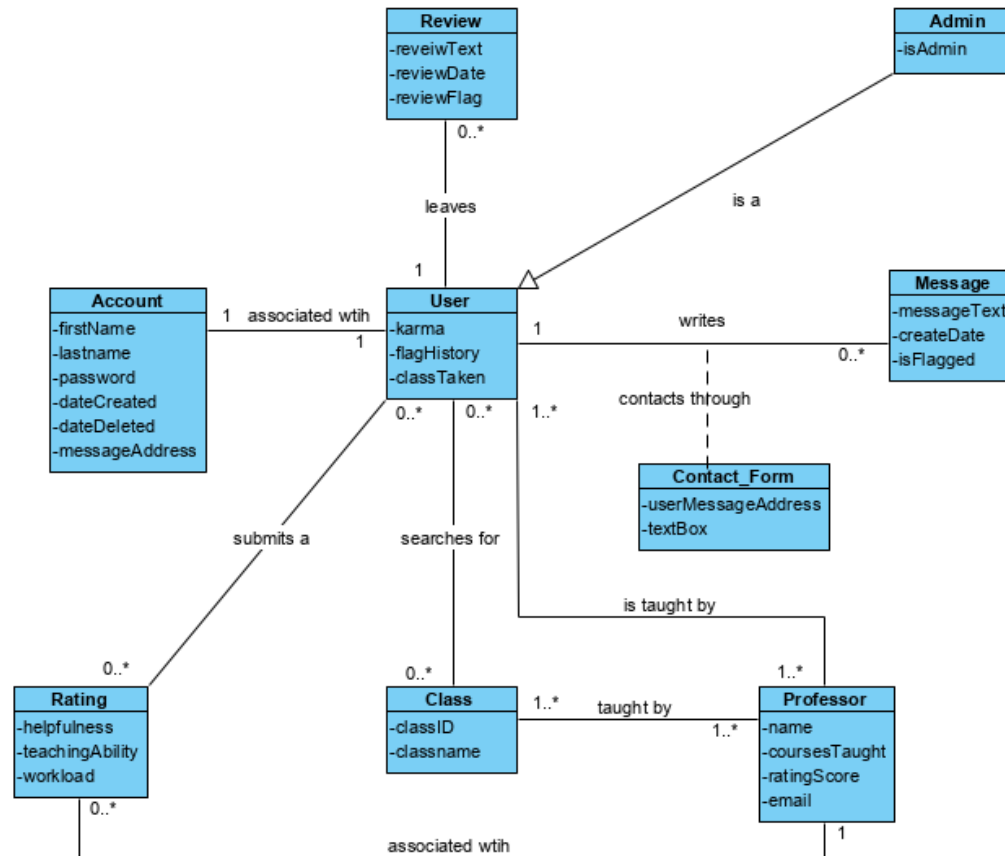
## Examples

- `checkUsername()`
- `checkPassword()`
- `populateClassList()`
- `populateProfessorList()`
- `sendMessage()`
- `viewMessage()`

There are ~70 more 😊



# DOMAIN MODEL



# QUESTIONS?

# TIMECARD

## Mark Fuller

09/10 - 1 Hour Use Case & Requirements List  
09/11 - 1 Hour Use Case  
09/12 - 1 Hour Use Case  
09/13 - 1 Hour Use Case  
09/14 - 1 Hour Use Case  
09/15 - 1 Hour Use Case  
09/16 - 1 Hour Use Case  
09/17 - 1 Hour Use Case  
09/18 - 1 Hour Use Case  
09/19 - 1 Hour Use Case  
09/20 - Domain Model

## Connor Woodahl

09/10 - 1 Requirements List  
09/11 - 1 Wireframes  
09/12 - 1 Wireframes  
09/13 - 1 Wireframes  
09/14 - 1 Wireframes  
09/15 - 1 Wireframes  
09/16 - 1 Wireframes  
09/17 - 1 Wireframes  
09/18 - 1 Wireframes  
09/19 - 1 Use Case Diagram & Traceability Matrix  
09/20 - 1 Use Case Diagram & Domain Model  
09/21 - 1 Use Case Diagram

## Kevin Kulda

09/10 - 1 Requirements List  
09/11 -  
09/12 -  
09/13 -  
09/14 -  
09/15 -  
09/16 -  
09/17 -  
09/18 -  
09/19 - 6 SSD  
09/20 - 6 Domain Model & System Operations  
09/21 - Presentation