Project: CSE 360 Help System Phase 3

Alan Lintemuth, William McLean, Max Neville, Tushar Sachan, Taj Yoshimura

CSE 360: Thursday Group 47

**Phase One Project Overview**

Everyone struggles with finding the help they need to excel in life; few struggle as intensely as undergraduate students. As a remedy, our team is building a help system specifically designed for Computer Science and Engineering 360 students at Arizona State University. This system will leverage questions and answers from previous years' Education Discussion boards to fill out a computer application to connect students with the information they need. Students will benefit from personalized settings and the ability to seamlessly change roles depending on what information they would like to access. Instructors will benefit from a system that allows them to manage and assist students effectively. The system will be coded in the well-known coding language, Java, utilizing JavaFX to add easy to use graphical features. Our phase one deliverables will include all the methods necessary to build out our help system in the next phases.

In this system we have specified 3 specific roles of users: admins, students, and instructors. The role of admin has the ability to perform certain actions that the other user types cannot (though the role admin may also be possessed at the same time as the student or instructor role). These specific admin abilities are to:

* Invite someone to join the application by giving a new user a one-time password
* Reset a user account
* Delete a user account
* List the user accounts showing each user with their associated names and role(s)
* Remove or add a role from a user
* Log out

This differs from the abilities we give to all other users, students or instructors, which are to:

* Create an account from an invitation given by an admin utilizing a one-time password given from an admin, allowing a user to create an account with its own unique password
* Select their role: Admin, Instructor, and/or Student as allowed by an admin
* Log out

A major focus within this phase has been the security of the system. In order to keep our program, secure we added multiple measures such as a system of one-time passwords with expiration dates that are used within the process of creating new accounts, and keeping passwords protected by storing them as a more secure data type. By creating an enterprise-level help application, our team will be investing in the success of future students at ASU. With the tools generated in phase one, we will create a robust platform to build all future deliverables. We aim to create a private, secure, efficient platform that anyone can leverage to help succeed in computer science.

**Phase Two Project Overview:**

In this project phase, we will leverage the previously written code to create a database of help articles to be accessed by the admins and instructors. These roles will be able to interact with these individual help articles. This phase entirely focuses on implementing the article database and its addition to the overall system, with plans for further refinement and utilization in the future. After this phase, users of our system will be able to add and save completed articles to the database; these articles will later be accessed by all help system users to facilitate greater learning.

The articles within the database will each contain a unique ID to identify them alongside a title, authors, a set of keywords, a body, a level indication, a short description, references, and groups. Admins and instructors in the database will have the following abilities when it comes to interaction with the article database:

* Adding, Updating, Viewing, and Removing an article
* Listing articles (by group or as a whole)
* Backup articles to a file of the user’s specification (by group or as a whole)
* Restore articles from a file (by merging the restored file with the existing database or replacing it entirely)

These functions will allow admins and instructors to efficiently create and operate the database. Groups of articles can be created, backed up, and restored so that each class at ASU can have the most pertinent help for the semester. Admins and instructors having the ability to list all the groups will make it easy to select from these slates. An additional layer of security is added in the form of encrypting the body of each article. When backed up to a file, the contents of the article’s body will be encrypted. The body is only readable when the admin/instructor chooses to list the articles. All of this new functionality will build out our platform, getting closer to our goal of an efficient help system.

**Change Bars: Project Phase Three**

**Our project is nearing completion as we complete the penultimate stage of development. This stage focuses on security for both users and instructors in the system. Moving forward, we will ensure all passwords are encrypted to keep all user accounts secure. A new class and group of articles is being created with encrypted bodies to keep proprietary information secure as well. Instructors can add users to these groups to allow viewing of decrypted articles. The student class is being added to allow a whole new class of user access to the help system. Students will be able to:**

* **Send help queries to system administrators**
* **Search for relevant articles to answer their class questions**
* **Organize these searches by difficulty level and group designation**
* **Ask for access to special groups with encrypted articles**

**Instructors will also be gaining new functionality to keep their information secure and allow easy creation of class groupings. They will be able to:**

* **Create general groups for anyone to view articles on the system**
* **Create specialized groups to keep information confidential**
* **Create, view, edit, or delete articles and groups of articles**
* **Add, view, and delete students from the help system and groups**

**This new functionality will allow group and article creation for any college class, research unit, and/or curriculum team. The administrator class will also be refined, requiring them to be added explicitly to groups while preserving most of their other functions. These additions will create the framework for our final secure product. After a few additional refinements in the next phase, our help system will be operational for any school needing a safe, searchable database of help for students.**

**Phase 1 Requirements and User Stories:**

We will be formatting our User Stories in the following manner: As a *role* I want to *action* so that *benefit*

As a *user,* I want to *get help with CSE 360* so that *I can succeed in Computer Science*.

As a *new user*, I want to *set up my account properly so* that *my information will be stored correctly.*

As a *user*, I want to *be assigned the proper role so* that *I can access the information that most pertain to my questions.*

As a *user*, I want to *be able to switch roles* so that *I can access the correct information for my session.*

As a *user,* I want to *have a secure password* so that *my credentials will not be stolen*.

As *a user,* I want to *easily* *view my homepage so* that *I can gain an understanding of the help system*.

As a *user*, I want to *be able to log out* so that I *can finish my session*

As an *administrator*, I want to *establish an account to manage the system database* so that *it will be properly operated.*

As an *administrator*, I want *the first person to use the system to get an admin account* so that *the system will always have an administrator*.

As an *administrator*, I want *to invite people to join my application* so that *they can use my help system*.

As an *administrator*, I want to *choose the role for people when I send an invitation out* so that *they can assume the proper role.*

As an *administrator*, I want *to reset user accounts* so that *I can fix issues with their accounts.*

As an *administrator*, I want *to delete user accounts* so that *the database can function efficiently*.

As an *administrator*, I want to *be able to see all of the user accounts* so that *I can see who is using the application*.

As an *administrator*, I want to *change the roles of people using the help system* so that *I can keep users organized*.

Distilling down these user stories into Phase One Requirements gives us the best opportunity to focus on the needs of all stakeholders. Creating multiple user roles (Admin, Student, and Instructor) with corresponding home pages will be required to facilitate help system use and management. Data will have to be stored and utilized to organize the system. The system requires an administrator who can invite users, manage accounts, manage the system, and ensure all information is handled securely. Lastly, each user will require some functionality on their homepage to begin navigating the help system. By including all these requirements in Phase One, we will create a solid base to build out our ASU Student Help System.

**Phase Two Requirements and User Stories:**

As an *administrator*/*instructor*, I want to *be able to create articles* so that *I can add articles to the database.*

As an *administrator*/*instructor*, I want to *be able to add relevant information (such as titles, descriptions, bodies, etc.) to articles* so that *I can create meaningful help articles for students and instructors.*

As an *administrator*, I want to *ensure each article has a unique entry* so that *there will be no duplicate articles.*

As an *administrator*/*instructor*, I want to *be able to update, view, and delete articles* so that *I can modify articles and keep them valid.*

As an *administrator*/*instructor*, I want to *be able to backup and restore the articles* so that *I can keep the correct database.*

As an *administrator*/*instructor*, I want to *be able to create a separate backup file* so that *I can save a backup file safely.*

As an *administrator*/*instructor*, I want to *be able to create and backup a group of articles* so that *I can easily organize the help topics.*

As an *administrator*/*instructor*, I want to *be able to list all of the articles and groups of articles* so that *I can easily organize the help topics.*

The user stories of this phase represent the addition of the help articles and their functionality. Creating new articles with the correct fields is important to instructors and students seeking help. The ability to group articles by class or subject will streamline connections between users. Backing up and restoring the database and groups will ensure minimal interruption. The functionality added in this phase gives us a solid foundation for an efficient help system based on articles uploaded by admins and instructors.

**Change Bars: Project Phase Three**

**As a *student*, I want to *be able to send messages to administrators/instructors* so that *I can get help with the help system articles.***

**As a *student*, I want to *search for help articles* so that *I can find articles to assist me in my assignments.***

**As a *student*, I want to *narrow down the difficulty level of articles I am searching for* so that *my search will efficiently return the articles I need.***

**As a *student*, I want to *specify the group of articles I am searching for* so that *my search will return the correct articles.***

**As a *student*, I want to *search the articles in the group of articles I am searching for* so that *my search will return the correct articles.***

**As a *student*, I want to *have the search articles displayed efficiently* so that *choosing the right article will be easy.***

**As an *instructor*, I want to *be able to view who has special permissions for a group* so that *I will know who to reach out to for access.***

**As an *instructor*, I want to *be able to view which students have special permissions for a group* so that *I will know who has access and who needs it.***

**As an *instructor*, I want to *be able to search just like the students* so that *I will know which articles they have access to.***

**As an *instructor*, I want to *be able to create, view, edit, and delete special group articles* so that *I can control access to proprietary information.***

**As an *instructor*, I want to *be able to add students to special groups* so that *they can* *access select articles with permission.***

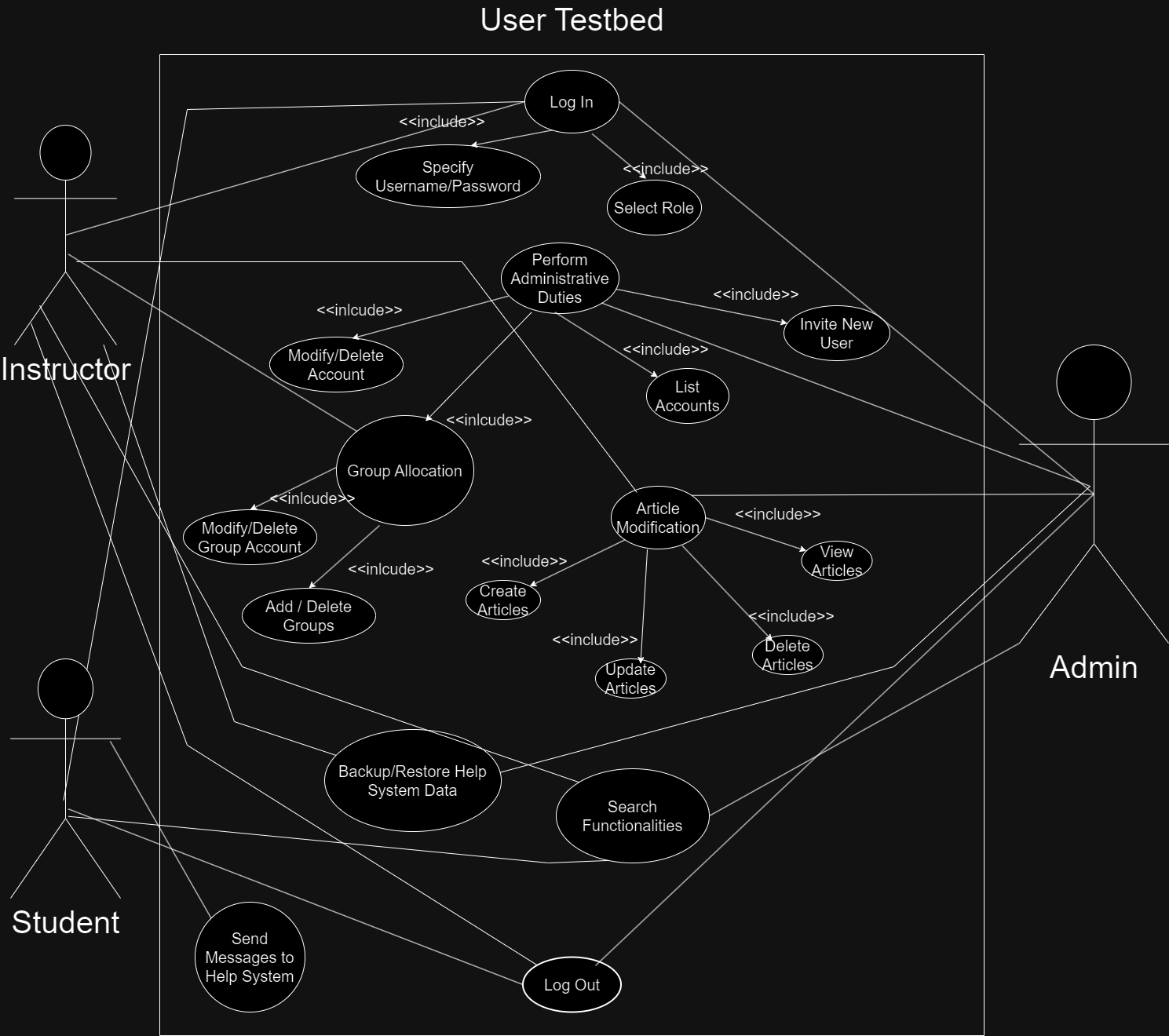
**As an *instructor*, I want to *be able to remove users from special groups* so that *I can* *control who has access to the system.***

**As an *administrator*, I want to *control general groups and which users are associated with them* so that *the system can be run efficiently.***

**As an *administrator*, I want to *be able to receive special permission for a group* so that *I can help instructors manage their groups.***

**This third phase creates protection for proprietary information within the help system. Instructors will be able to create encrypted articles and store them in special access groups. They can then add other users to their groups, ensuring specific special access articles can be viewed. Different instructors and administrators can be added to these special groups, while students, administrators, and instructors can be deleted. The student role is being created with the utility to send help messages, search for articles, view general articles, and view encrypted articles with special access. This section sets up our help system to become fully operational, with only one more phase of refinements.**

**Use Case:**

****

**UML:**

**Class Responsibility Collaborator:**

|  |  |
| --- | --- |
| CSE 360 Help System |  |
| Actors | Administrators, Instructors, Students, Help System, Help System Database |
| Description | An administrator will carry out all the functions of the database (inviting new users, resetting accounts, deleting accounts, changing user roles). Instructors and Students can sign up for an account, enter their information and password, and logout.  Admins and Instructors can create, update, view, and delete help articles. Admins and Instructors can create and backup groups of articles, and list all the articles and subgroups. |
| User Data | Each actor’s userID, password, first, middle, last, preferred name, and their role. |
| Article Data | Each article’s header, title, authors, abstract, keywords, body of the article, group. |
| Stimulus | Invitation issued by an administrator, or a user logging into their account. Admin or Instructor adding and manipulating articles in the database; adding and manipulating groups of articles. |
| Response | System asks for a one-time login password, then accepts a new password. Conformation of logout. System asks for article data, group data, and backup/restore data. |
| Comments | Users must be specifically invited by an administrator. Their invitation comes with a one-time password that will expire. Two database tables are created to store user data and article data. Database tables can be backed up and restored from files. |

|  |  |
| --- | --- |
| Home  This JavaFX class is the entry point to our GUI and calls methods to login, sign up, and handle admin sign in | Collaboration |
| Firstadminlogin(VBox, StartCSE360) | DatabaseHelper |
| usercreation(VBox, StartCSE360) | Roles |
| userlogin(VBox, StartCSE360) |  |
| setrole(String, String): void | Roles |
| userdetail(VBox, , String, StartCSE360) |  |
| String rolehelper(Boolean(B), B, B) |  |
| loginUserRoles((VBox, StartCSE360, String) |  |

|  |  |
| --- | --- |
| DatabaseHelper  This class deal with the data moving to and from the database | Collaboration |
| role (String) | StartCSE360 |
| register(String, String, String) | StartCSE360 |
| update(String(S), S, S, S, S) | StartCSE360 |
| displayUsers | StartCSE360 |
| deleteuser(String) | StartCSE360 |
| setUsers | StartCSE360 |

|  |  |
| --- | --- |
| ArticleDatabaseHelper  This class deal with the data interfacing with the articles database | Collaboration |
| connectToDatabase() | articlehelper |
| mergeArticles(string) | articlehelper |
| viewArticle(long) | articlehelper |
| addArticle(S, S, S, S, S, S, S, S) | articlehelper |
| addRestoredArticle(S, S, S, S, S, S, S, S) | articlehelper |
| deleteArticle(String) | articlehelper |
| listArticles() | articlehelper |
| listGroupedArticles() | articlehelper |
| backupArticles(String) | articlehelper |
| backupGroupedArticles(String, String) | articlehelper |
| addRestoredArticle() | articlehelper |
|  |  |
| Roles  This class handles the roles that the person has | Collaboration |
| setrole(String) | Admin |
| administructor(Stage) | Instructor |
| adminstudent(Stage) | Student |
| admin(Stage) | Admin |
| logout(Stage) | Admin, Instructor, Student |
| instructor(Stage) | Instructor |
| student(Stage) | Student |

|  |  |
| --- | --- |
| Admin  This class handles the abilities of an admin | Collaboration |
| Showusers | Roles |
| removeroles(User, enum) | Roles |
| Invite | CSE360 |
| reset(User) | CSE360 |
| delete(User) | CSE360 |
| addRole(User, enum) | Roles |
| editroles(String, BorderPane) |  |
| setUsers |  |

|  |  |
| --- | --- |
| Users  This class handles the abilities of Users | Collaboration |
| Showusers | Roles |
| removeroles(User, enum) | Roles |
| Invite |  |
| reset(User) |  |
| delete(User) |  |
| addRole(User, enum) | Roles |
| editroles(String, BorderPane) |  |
| setUsers |  |

|  |  |
| --- | --- |
| articlehelper  This class is the front end for article database class and has functions to handle data | Collaboration |
| addarticle(Stage) | ArticleDatabaseHelper |
| remove(Stage) | ArticleDatabaseHelper |
| list(Stage) | ArticleDatabaseHelper |
| update(Stage, String, VBox, long) | ArticleDatabaseHelper |
| backup(Stage) | ArticleDatabaseHelper |
| backupall(Stage, VBox) | ArticleDatabaseHelper |
| backupgroups(Stage, VBox) | ArticleDatabaseHelper |
| restore(Stage) | ArticleDatabaseHelper |
| restoreall(Stage, VBox) | ArticleDatabaseHelper |
| restoremerge(Stage, VBox) | ArticleDatabaseHelper |
| searching(Stage) | ArticleDatabaseHelper |
| searchbyid(Stage, VBox) | ArticleDatabaseHelper |
| searchbygroups(Stage, VBox) | ArticleDatabaseHelper |
| printgroups(Stage, VBox, String | ArticleDatabaseHelper |

|  |  |
| --- | --- |
| StartCSE360  This class is the front end for the database class and has necessary functions to handle data | Collaboration |
| setupAdministartor(String, String, String): int | DatabaseHelper |
| userSetUp(String, String, String): int | DatabaseHelper |
| prefname(String): String | DatabaseHelper |
| delete(String): void | DatabaseHelper |
| findrole(String): String | DatabaseHelper |
| login(String, String): Boolean | DatabaseHelper |

**Screencasts:**

**Phase 3 Technical Screencast:**

Phase 2 Technical Screencast: [https://asu.zoom.us/rec/share/tXCMb4Q2go\_zI77TvtCsA\_Nz7nt6YCSQU2NKuYZlmDz9Yq-CBytZyRZ5KugnYK6.mfhjFjAkhsVkHo8C](https://asu.zoom.us/rec/share/tXCMb4Q2go_zI77TvtCsA_Nz7nt6YCSQU2NKuYZlmDz9Yq-CBytZyRZ5KugnYK6.mfhjFjAkhsVkHo8C%20)

Passcode: Q2Z88OF^

Phase 1 Technical Screencast:

<https://asu.zoom.us/rec/share/4xoL7wsDhevcYB-IWa99C2NkBXsQtZvDpQflO1E0WXQJ3YtOpHJim5KisIek5KHR.vHLLVy6AiRU9ZLNe?startTime=1728542604000>

Passcode: P00Y%6r@

**Phase 3 Technical Screencast:**

Phase 2 Demo Screencast: [https://asu.zoom.us/rec/share/4AlwFvs8IUCLut2gvGcF\_\_85y08RZYKTZQld5AYhNmd3L04VXLgrKvk KoH3UnZ2Y.yVdixQyWYT1B14\_L?startTime=1730355374000](https://asu.zoom.us/rec/share/4AlwFvs8IUCLut2gvGcF__85y08RZYKTZQld5AYhNmd3L04VXLgrKvk%20KoH3UnZ2Y.yVdixQyWYT1B14_L?startTime=1730355374000%20)

Passcode: q+xxbiP2

Phase 1 Demo Screencast:

<https://asu.zoom.us/rec/share/m3yLcQi2a2pInwLi-mbtU-VGVfJ6OiiK5yhC1N9HBhvFVPR2Nu1bIY7p_2I4fSTI.hhEOIhqZ_Km8LJ4y?startTime=1728537570000> Passcode: n!FB03%X

**GitHub Link:**

<https://github.com/mnevi/CSE360Project.git>

Credit Page

|  |  |
| --- | --- |
| Team Member Name | Contributions |
| Alan Lintemuth | Oversaw majority of the documentation including the Project Overview and User Stories segments. Worked with detailed testing of the project using JUnit functionality. |
| William McLean | Moral support. |
| Tuschar Sarchan | Implemented the entire GUI aspect for group functionality across the updated Student and Instructor classes; and helped with integration with the set functionality. Helped with testing/error analysis. |
| Max Neville | Completed much of the design and architecture for the project,  creating the UML/Use Case diagrams. Also recorded both  screencasts. Helped with testing / error analysis / credit page. |
| Taj Yoshimura | Worked on programming group functionality across the various classes, primarily working with the newly implemented group database. Also helped to create the CRC part of the document. Helped with testing / error analysis. |