Detailed Design Model

for

SAMS 2024

Version/Iteration 2.0 approved

Prepared by Group 2
Londhe, Soudagar, srl1622
Sajjala, Swetha, ss5743
Skalicky, Stacy, sms7705
Vellaisamy Senthilkumar, Rohini, rv8542

SWEN-746
Rochester Institute of Technology

05 December, 2023

Document Deliverables

This document presents the Detailed Design Model, offering a comprehensive and in-depth overview of the system's architecture, components, and their interrelationships. Explore this detailed guide for a thorough understanding of the intricacies that underlie the development process. This document mainly covers the following deliverables:

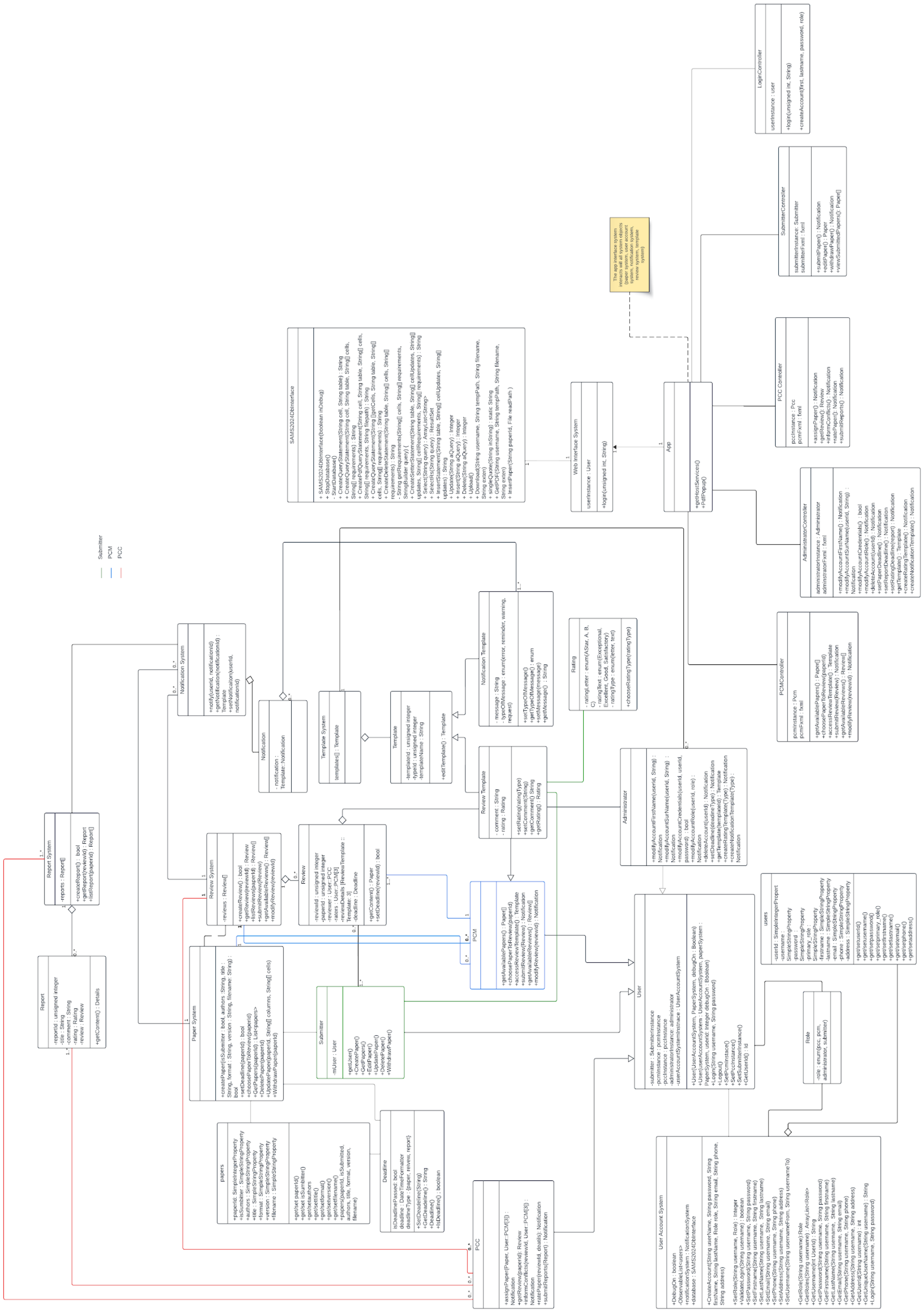
1. **Fully Detailed Class Diagram**
   The fully detailed class diagram is an iterative refinement of our previous domain diagram, systematically incorporating additional attributes and methods. This enhancement ensures a seamless transition to code implementation, providing a more thorough representation of the system's structure and functionalities.
2. **Behavioral Diagrams**
   The behavioral diagrams in this document encompass both a ***State Diagram***, offering an overarching view of the application's flow, and ***Object Sequence Diagrams***. These object sequence diagrams represent an iteration of our initial sequences, now incorporating actual system calls and interactions through a web interface. Together, they provide a comprehensive understanding of the dynamic aspects of the system's behavior.
3. **Implementation Decisions**
   With the completion of the implementation phase by the team, the reiterated document now includes detailed insights into decisions made throughout the implementation process. It presents the user with a streamlined flow of the main flow, complemented by illustrative screenshots, providing a comprehensive overview of the application's functionality and design.

This document is the result of iterative refinement over our following preceding design models, ensuring a meticulous and coherent evolution – each iteration has contributed to the development of a comprehensive and optimized Detailed Design Model:

- SAMS 2024 Details Requirements Model using Use Cases
- SAMS 2024 Actor/System Sequence Model
- SAMS 2024 Domain Model
- SAMS 2024 Architectural Model

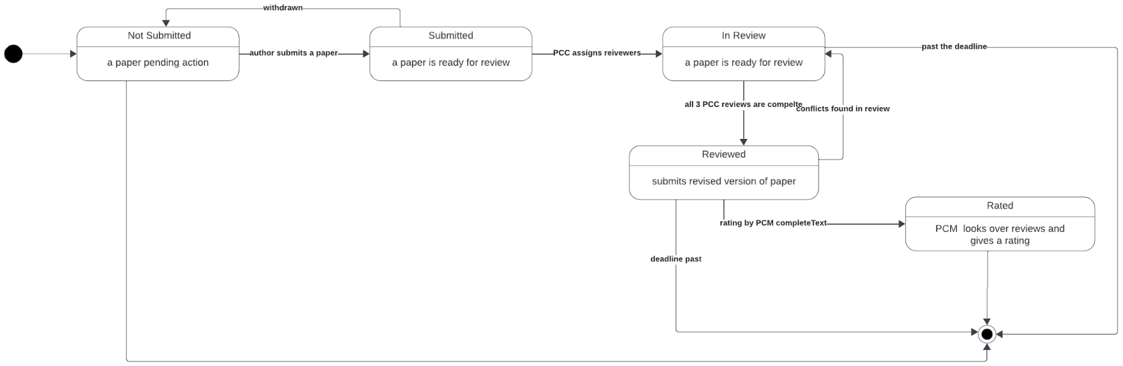
1. **(Fully Detailed) Class Diagram**

The following detailed class diagram depicts a comprehensive blueprint for software development, showcasing a clear and intuitive representation of the system's architecture and components.



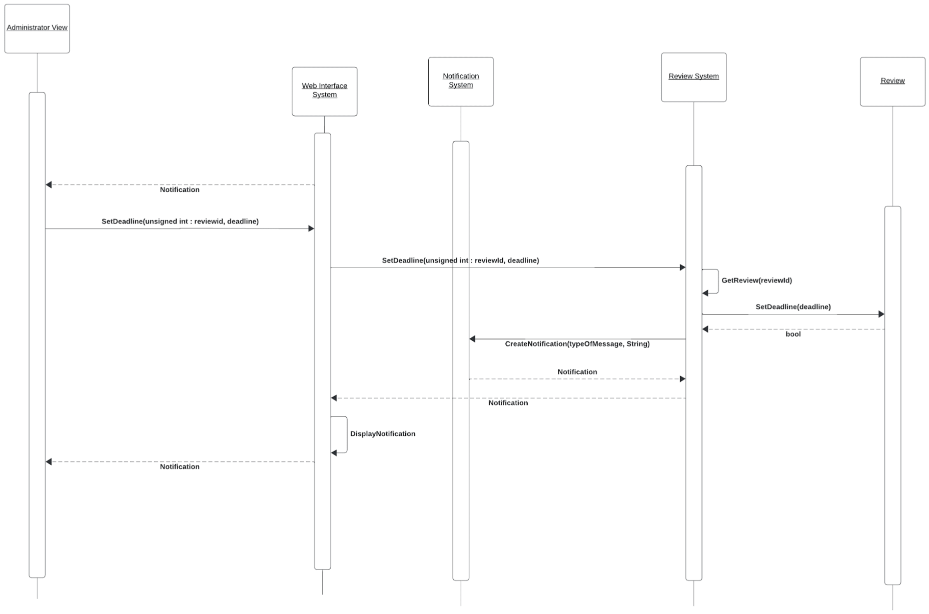
2. Behavioral Diagrams

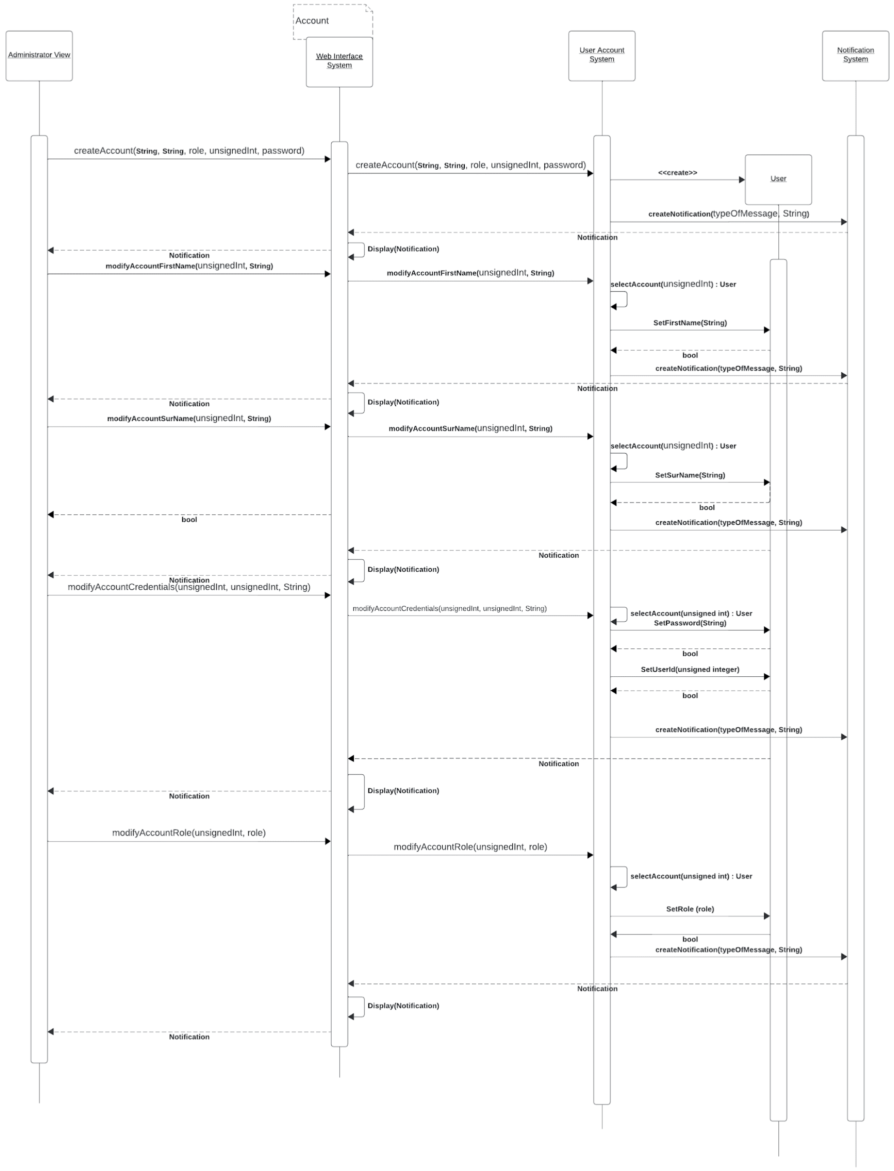
State Diagram: The following diagram illustrates a state machine diagram, representing the various states and transitions of an object or system in response to events. This state diagram demonstrates the transition of a paper as it is evaluated for the conference. Not submitted, is before the paper has been indicated by a submitter to be submitted. Submitted is a state that is ready for PCC allocation for review. In review is a state awaiting evaluation from at least 3 PCCs, once complete the paper transitions to review the paper is rated by a PCM and the cycle completes. Should the submitter withdraw before the paper transitions to in review, the paper stays in not submitted until submission is initiated again. If the deadline passes while not submitted, in review or not yet rated the cycle ends and the paper does not follow on to other states and remains in the state left off before deadline passed.

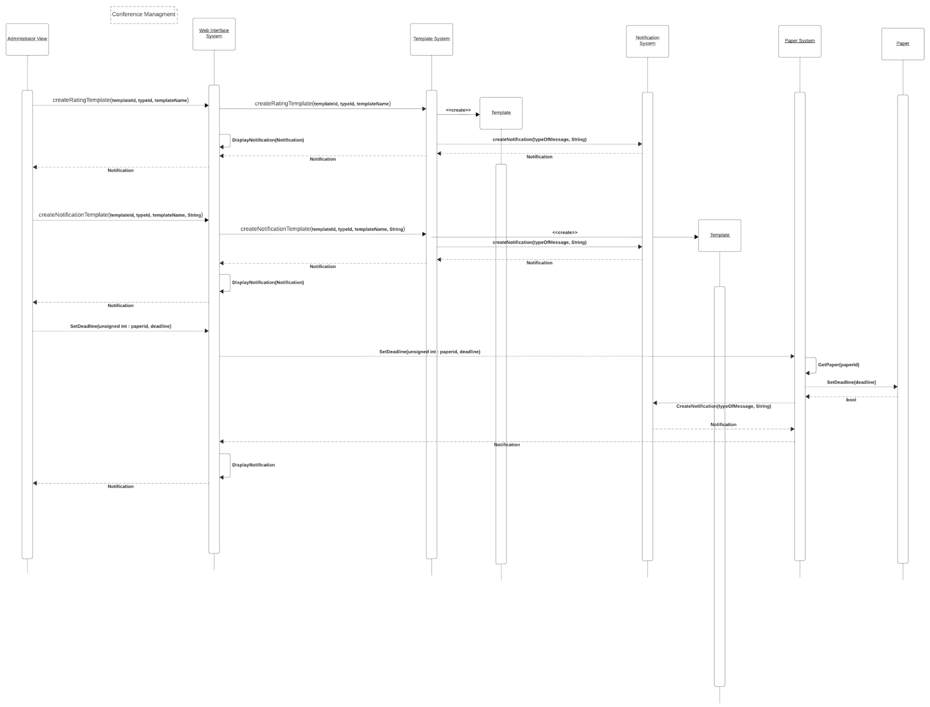

withdrawn
Not Submitted
a paper pending action
author submits a paper
Submitted
a paper is ready for review
PCC assigns reivewers
In Review
a paper is ready for review
past the deadline
all 3 PCC reviews are compelte
conflicts found in review
Reviewed
submits revised version of paper
rating by PCM completeText
Rated
PCM looks over reviews and gives a rating
deadline past

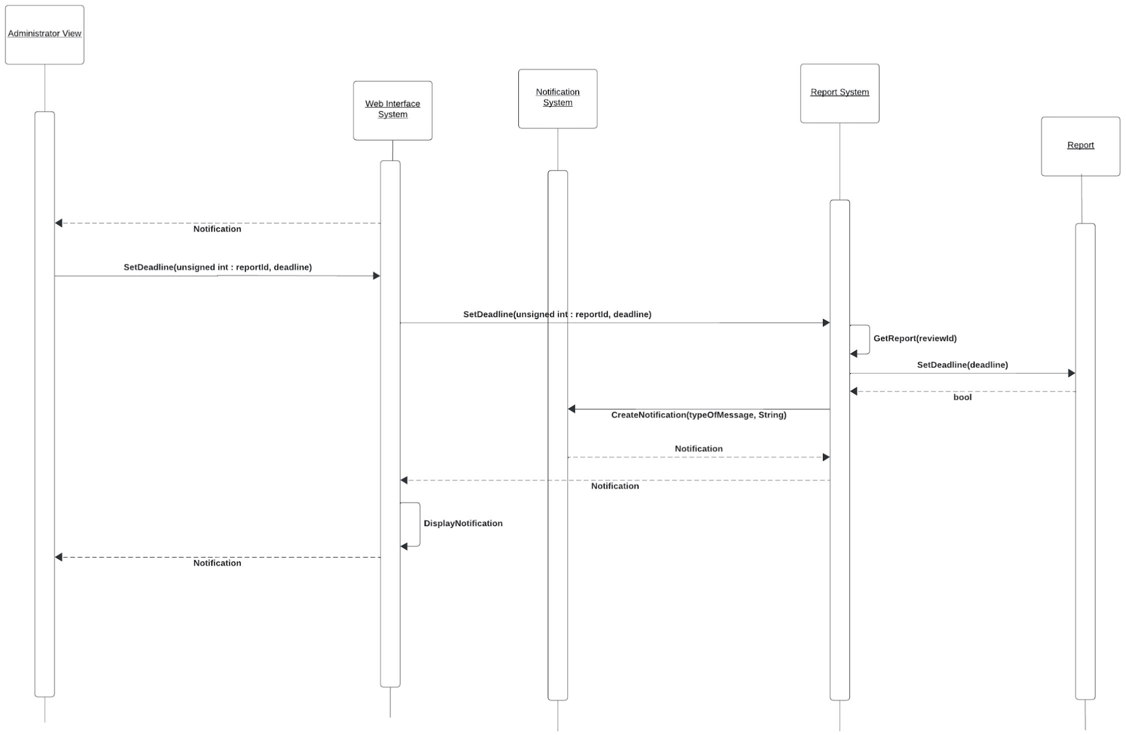
Object Sequence Diagram: The following object sequence diagram(s) illustrate the interactions and messages exchanged between objects in a particular scenario or during a specific process flow.

Administrator: There are following 4 sequence diagrams that detail the account management, template management and deadline management in that order.

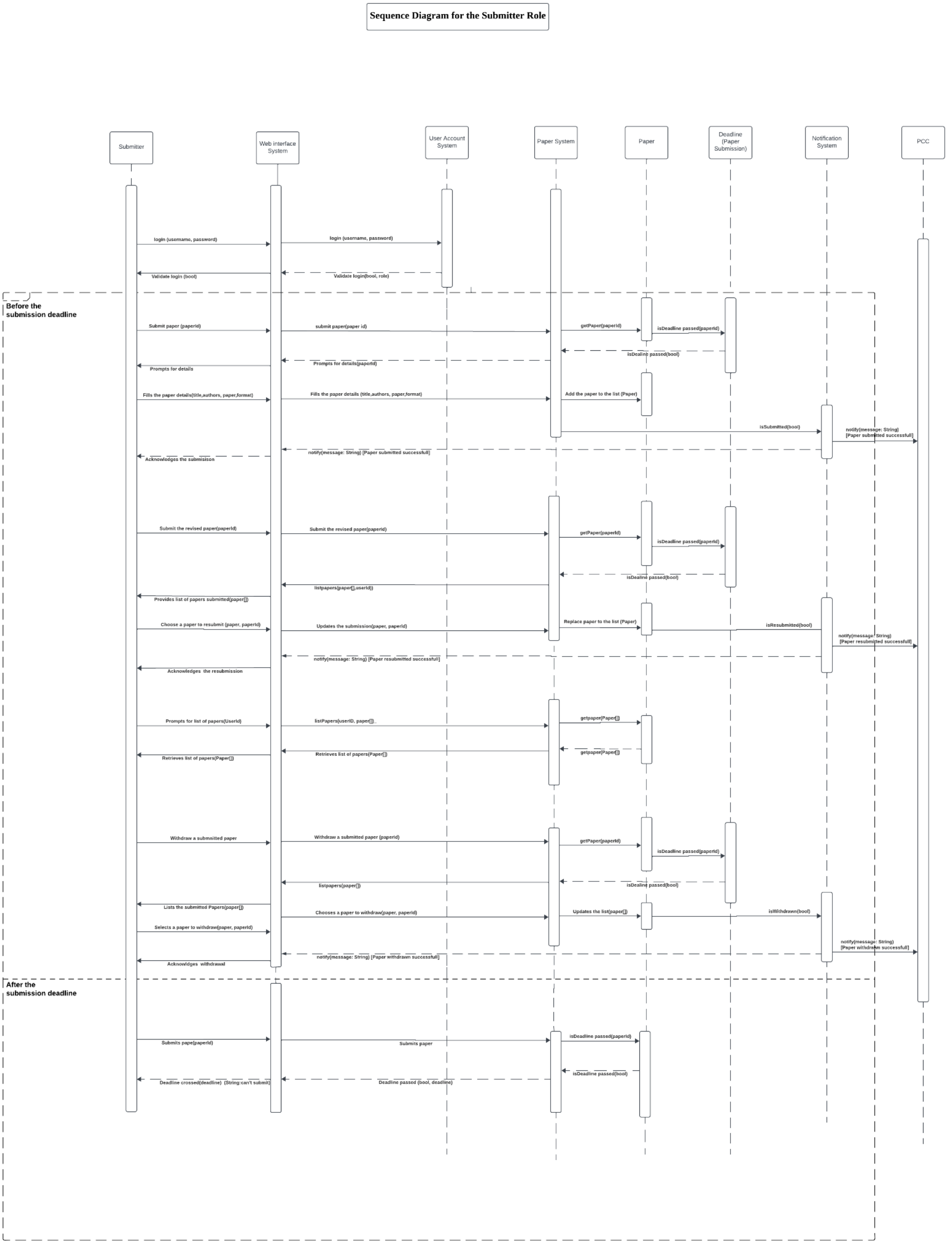



Account
Administrator View
Web Interface System
User Account System
Notification System
createAccount(String, String, role, unsignedInt, password)
createAccount(String, String, role, unsignedInt, password)
<<create>>
User
createNotification(typeOfMessage, String)
Notification
Display(Notification)
Notification
modifyAccountFirstName(unsignedInt, String)
modifyAccountFirstName(unsignedInt, String)
selectAccount(unsignedInt) : User
SetFirstName(String)
bool
createNotification(typeOfMessage, String)
Notification
Display(Notification)
Notification
modifyAccountSurName(unsignedInt, String)
modifyAccountSurName(unsignedInt, String)
selectAccount(unsignedInt) : User
SetSurName(String)
bool
bool
createNotification(typeOfMessage, String)
Notification
Display(Notification)
Notification
modifyAccountCredentials(unsignedInt, unsignedInt, String)
modifyAccountCredentials(unsignedInt, unsignedInt, String)
selectAccount(unsigned int) : User
SetPassword(String)
bool
SetUserId(unsigned integer)
bool
createNotification(typeOfMessage, String)
Notification
Display(Notification)
Notification
modifyAccountRole(unsignedInt, role)
modifyAccountRole(unsignedInt, role)
selectAccount(unsigned int) : User
SetRole (role)
bool
createNotification(typeOfMessage, String)
Notification
Display(Notification)
Notification

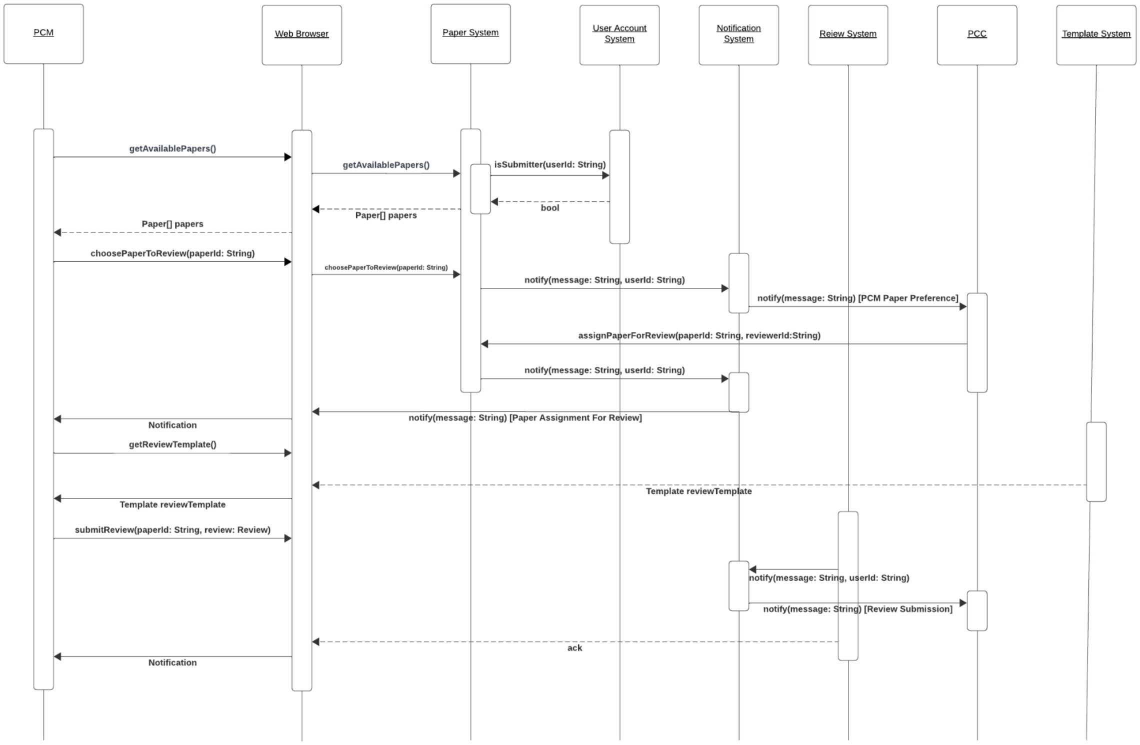

Conference Managment
Administrator View
Web Interface System
Template System
Notification System
Paper System
Paper
createRatingTemplate(templateId, typeId, templateName)
createRatingTemplate(templateId, typeId, templateName)
<<create>>
Template
createNotification(typeOfMessage, String)
DisplayNotification(Notification)
Notification
Notification
Notification
createNotificationTemplate(templateId, typeId, templateName, String)
createNotificationTemplate(templateId, typeId, templateName, String)
<<create>>
Template
createNotification(typeOfMessage, String)
Notification
Notification
DisplayNotification(Notification)
Notification
SetDeadline(unsigned int : paperid, deadline)
SetDeadline(unsigned int : paperid, deadline)
GetPaper(paperId)
SetDeadline(deadline)
bool
CreateNotification(typeOfMessage, String)
Notification
Notification
DisplayNotification
Notification


Administrator View
Web Interface System
Notification System
Report System
Report
Notification
SetDeadline(unsigned int : reportId, deadline)
SetDeadline(unsigned int : reportId, deadline)
GetReport(reviewId)
SetDeadline(deadline)
bool
CreateNotification(typeOfMessage, String)
Notification
Notification
DisplayNotification
Notification

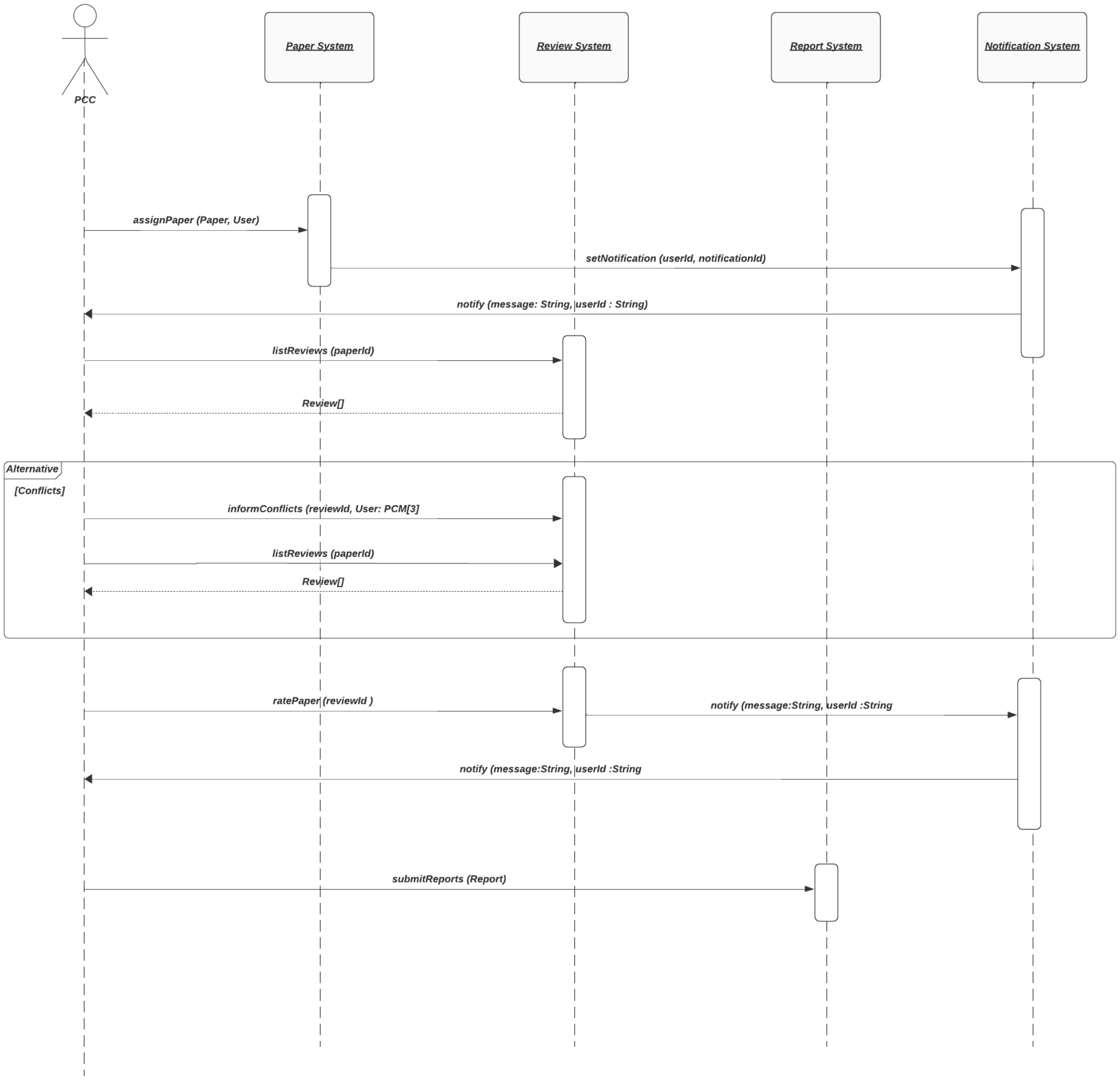
Submitter (Submit Paper):


Sequence Diagram for the Submitter Role
Submitter
Web interface System
User Account System
Paper System
Paper
Deadline (Paper Submission)
Notification System
PCC
login (username, password)
login (username, password)
Validate login (bool)
Validate login(bool, role)
Before the submission deadline
Submit paper (paperId)
submit paper(paper id)
getPaper(paperId)
isDeadline passed(paperId)
isDealine passed(bool)
Prompts for details(paperId)
Prompts for details
Fills the paper details(title,authors, paper,format)
Fills the paper details (title,authors, paper,format)
Add the paper to the list (Paper)
isSubmitted(bool)
notify(message: String) [Paper submitted successfull]
notify(message: String) [Paper submitted successfull]
Acknowledges the submisison
Submit the revised paper(paperId)
Submit the revised paper(paperId)
getPaper(paperId)
isDeadline passed(paperId)
isDealine passed(bool)
listpapers(paper[],userId))
Provides list of papers submitted(paper[])
Choose a paper to resubmit (paper, paperId)
Updates the submission(paper, paperId)
Replace paper to the list (Paper)
isResubmitted(bool)
notify(message: String) [Paper resubmitted successfull]
notify(message: String) [Paper resubmitted successfull]
Acknowledges the resubmission
Prompts for list of papers(UserId)
listPapers(userID, paper[])_
getpaper(Paper[])
getpaper(Paper[])
Retrieves list of papers(Paper[])
Retrieves list of papers(Paper[])
Withdraw a submnitted paper
Withdraw a submitted paper (paperId)
getPaper(paperId)
isDeadline passed(paperId)
isDealine passed(bool)
listpapers(paper[])
Lists the submitted Papers(paper[])
Chooses a paper to withdraw(paper, paperId)
Updates the list(paper[])
isWithdrawn(bool)
Selects a paper to withdraw(paper, paperId)
notify(message: String) [Paper withdrawn successfull]
notify(message: String) [Paper withdrawn successfull]
Acknowldges withdrawal
After the submission deadline
Submits pape(paperId)
Submits paper
isDeadline passed(paperId)
isDeadline passed(bool)
Deadline crossed(deadline) (String:can't submit)
Deadline passed (bool, deadline)

PCM (Review Paper):


PCM
Web Browser
Paper System
User Account System
Notification System
Reiew System
PCC
Template System
getAvailablePapers()
getAvailablePapers()
isSubmitter(userId: String)
bool
Paper[] papers
Paper[] papers
choosePaperToReview(paperId: String)
choosePaperToReview(paperId: String)
notify(message: String, userId: String)
notify(message: String) [PCM Paper Preference]
assignPaperForReview(paperId: String, reviewerId:String)
notify(message: String, userId: String)
notify(message: String) [Paper Assignment For Review]
Notification
getReviewTemplate()
Template reviewTemplate
Template reviewTemplate
submitReview(paperId: String, review: Review)
notify(message: String, userId: String)
notify(message: String) [Review Submission]
ack
Notification

PCC (Generate Report/Rating):

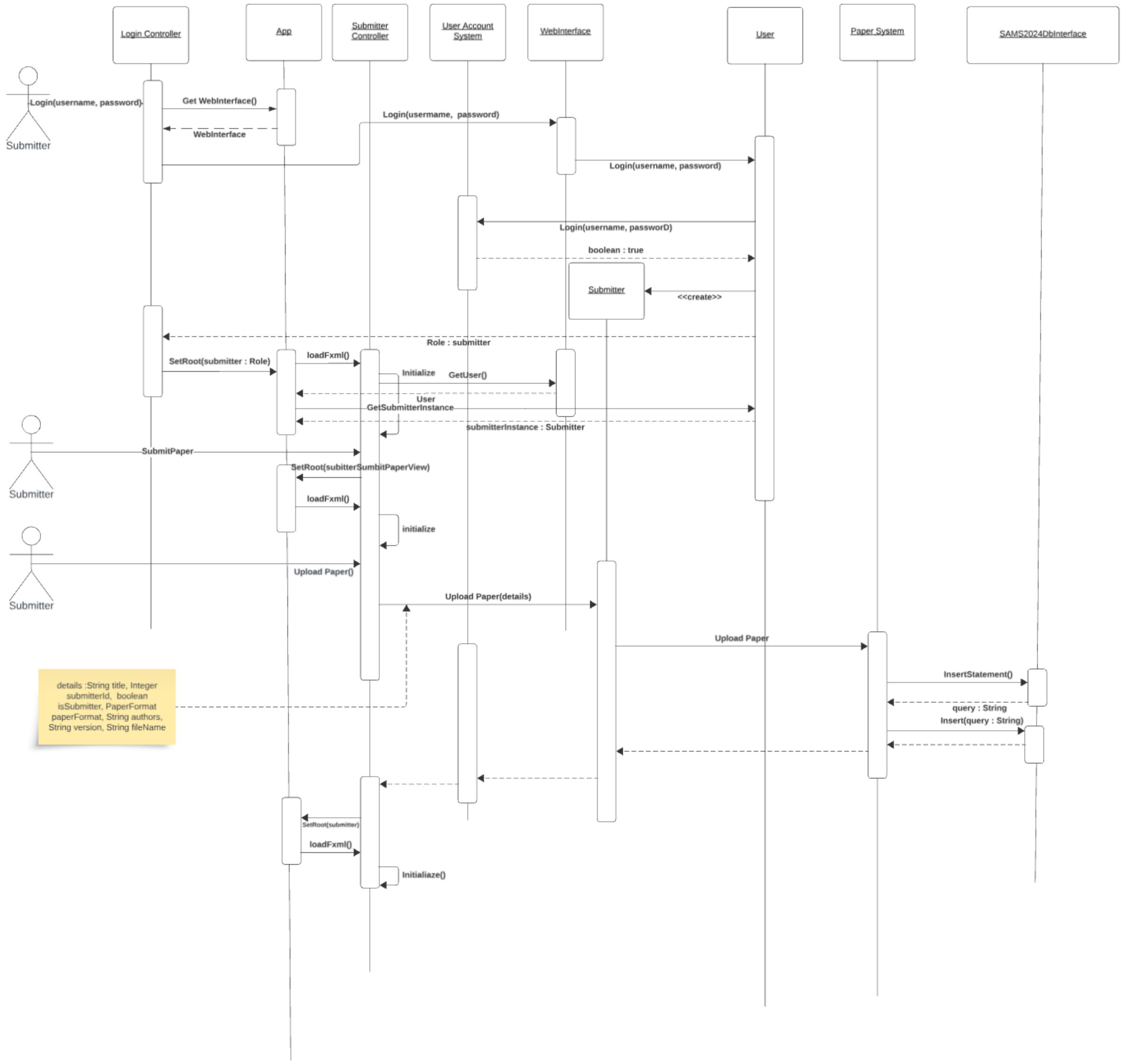

Paper System
Review System
Report System
Notification System
PCC
assignPaper (Paper, User)
setNotification (userId, notificationId)
notify (message: String, userId : String)
listReviews (paperId)
Review[]
Alternative
[Conflicts]
informConflicts (reviewId, User: PCM[3]
listReviews (paperId)
Review[]
ratePaper (reviewId)
notify (message:String, userId :String
notify (message:String, userId :String
submitReports (Report)

3. Implementation :

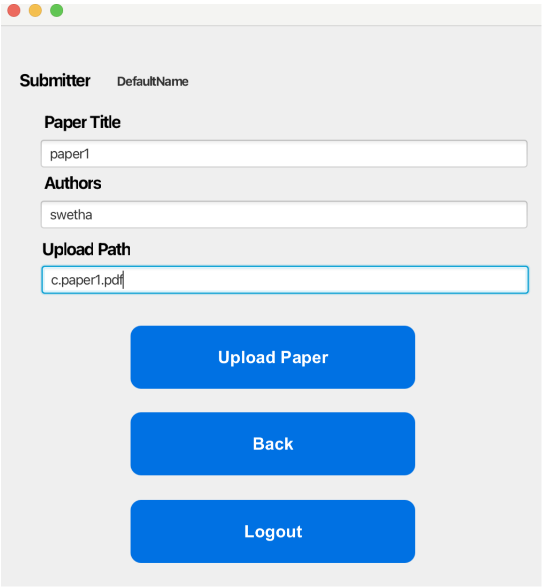
The implementation phase was executed in accordance with the models we created, ensuring a direct translation of our design concepts into functional and cohesive software. In terms of MVP, the team had agreed upon implementing the following two main flows in the SAMS 2024 system:

- **Submitter submitting a paper**
  In the main flow of a submitter submitting a paper, the process involves the submitter logging in, uploading a paper to the system, selecting the uploaded paper, and subsequently submitting it.

- Sequence Diagram for the main flow of submitter submitting the paper :



Screenshots of the application flow attached below:

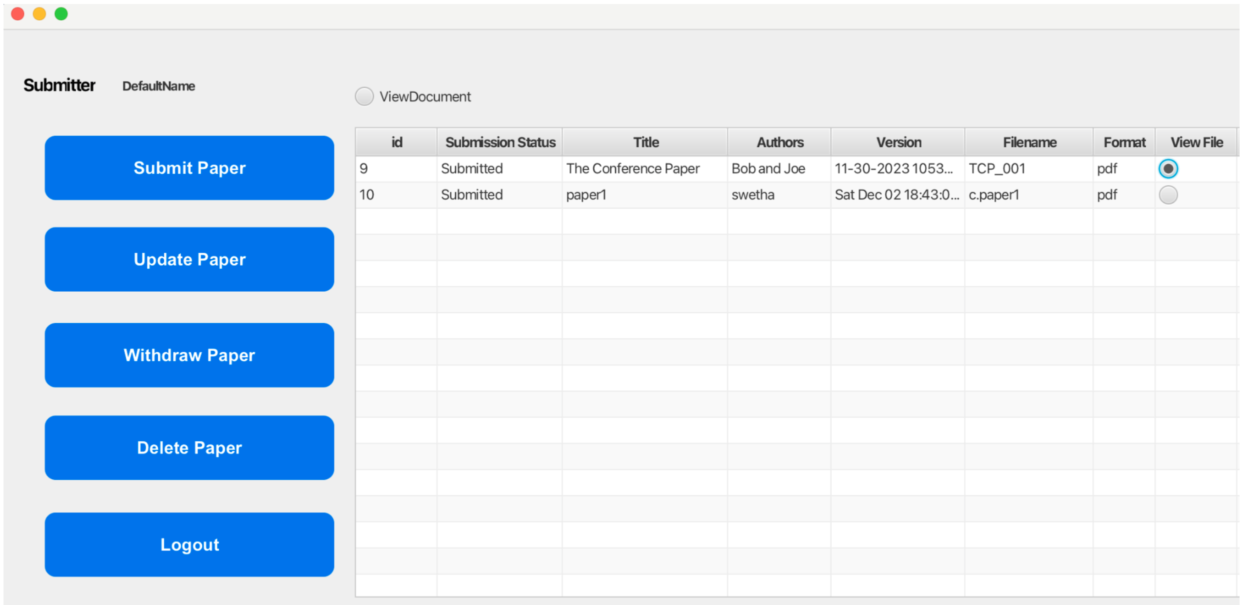


Submitter DefaultName

Paper Title: paper1

Authors: swetha

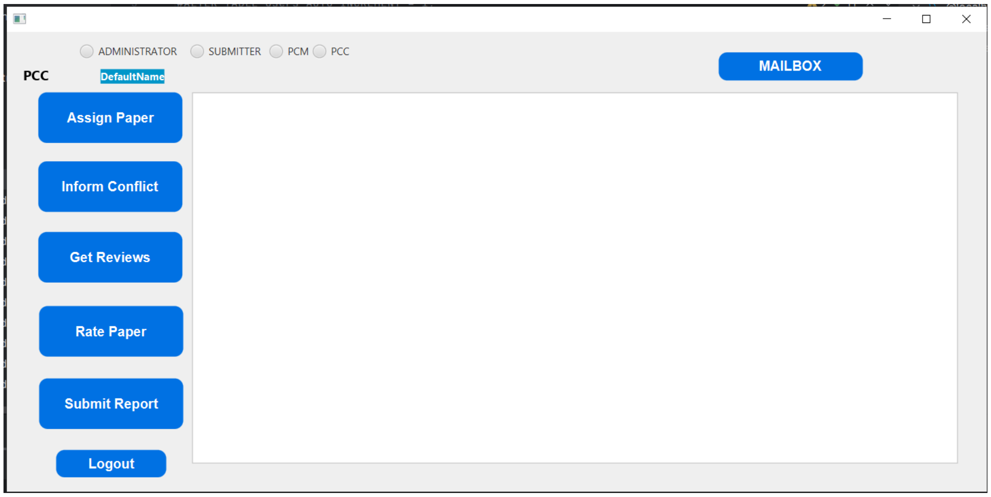
Upload Path: c.paper1.pdf

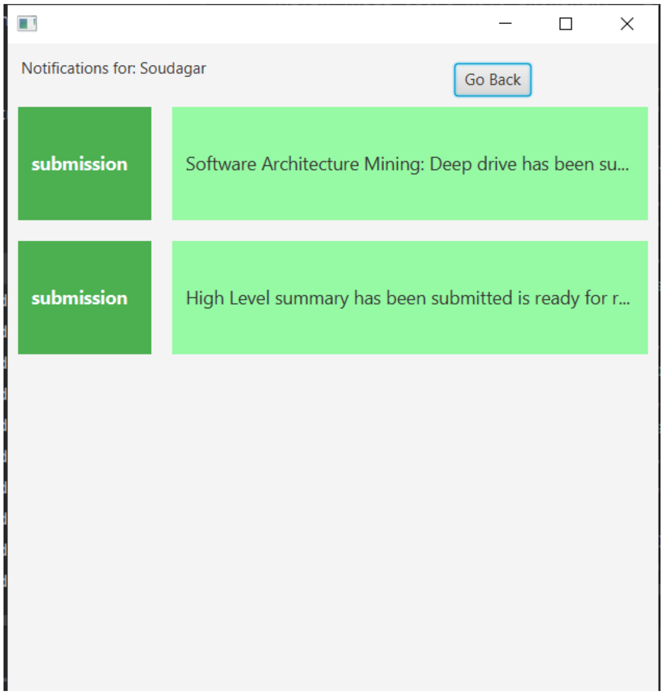


Submitter DefaultName

○ ViewDocument

id	Submission Status	Title	Authors	Version	Filename	Format	View File
9	Submitted	The Conference Paper	Bob and Joe	11-30-2023 1053...	TCP_001	pdf	●
10	Submitted	paper1	swetha	Sat Dec 02 18:43:0...	c.paper1	pdf	○





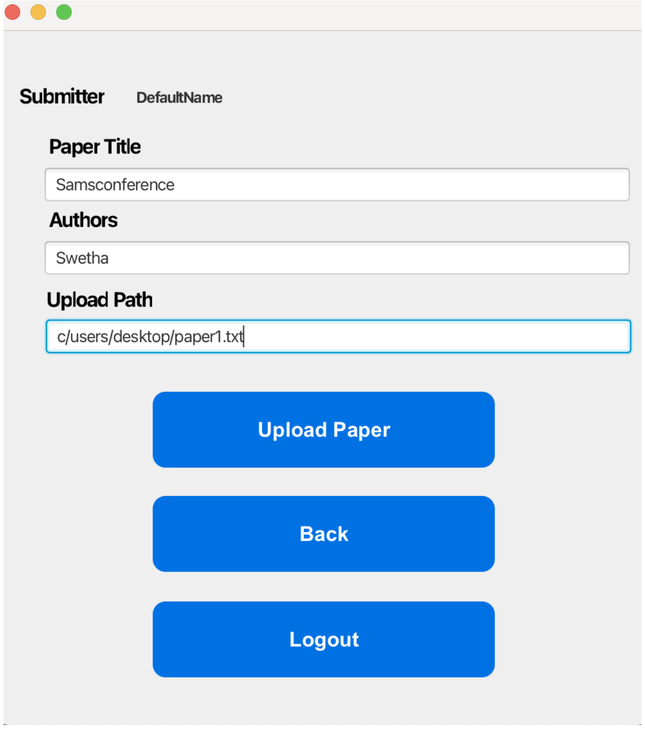
Notifications for: Soudagar

submission	Software Architecture Mining: Deep drive has been su...
submission	High Level summary has been submitted is ready for r...

After the Submitter submits the paper, PCC receives the notifications in his Mailbox once logged In. Notifications can be viewed in the form of title and messages which displays details for the author of submission, Title of the paper submitted and Timestamp on which the paper was submitted. After every new paper is submitted the mailbox for PCC gets populated.

The application also handles an error flow in the use case of submitting a paper to check the validity of the submitter paper (check on file type). For proper commitment to submission guidelines, this flow is only intended to accept documents in the formats of PDF or DOC. In case an author attempts to submit a paper in a format other than .pdf or .doc, the system immediately identifies the mistake and prevents the submission. The author is then directed to resubmit using the correct file type by the user interface, which clearly notifies them of the wrong format. This thorough validation process guarantees that authors can submit their articles without any difficulties while upholding the consistency and integrity of the submission format. It is combined with useful feedback and user-friendly cues.

Screenshots of the application flow attached below:



Submitter DefaultName

Paper Title

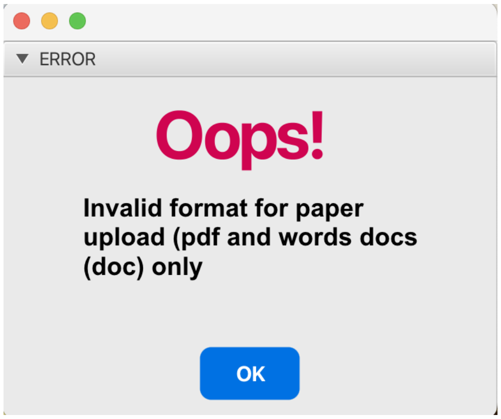
Samsconference

Authors

Swetha

Upload Path

c/users/desktop/paper1.txt

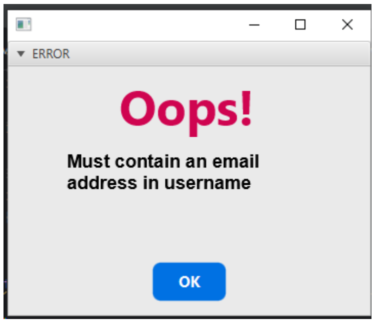


ERROR

Oops!

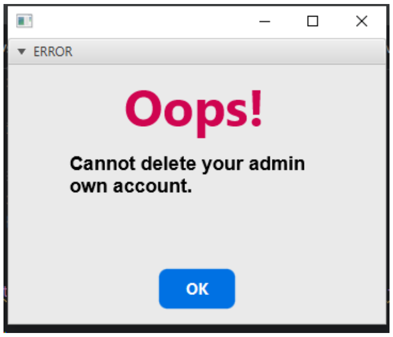
Invalid format for paper upload (pdf and words docs (doc) only

- **Administrator creating an account**
  The administrator possesses the capability to create user accounts with specified roles, facilitating precise control and assignment of roles within the system.
  Any user logged in as an administrator will be able to view all the same accounts. The mock screens for implementations was done in JavaFX hooking the database logic written in java to java that supports fxml/JavaFX display of screens. The functionality we were able to hookup was add account, delete account and modify account. The use case flows described in SAMS 2024 Details Requirements Model using Use Cases for Account Management was implemented. We were able to complete the main and alternative flows defined in the use case doc. We made the primary key the email so account creation expects an email with a format constrained to and at sign (@) and (.com) when an admin creates an account. Additionally, for delete accounts, the logged in administrator cannot delete themselves. When modifying an account the administrator can click into cells and edit content directly, however additional checking is still in place, changing of emails and roles must abide by the same constraints, only the 4 roles and only emails for username. Additionally, the phone number must be a 11 digit value. To assist the create account we added functionality so if a failure occurs the form refills so the administrator does not have to input it again. Some reaching steps we hoped to was to allow admin to switch between PCC, PCM and Submitter roles but did not get to hooking this up in time as shown by the radial buttons on the admin page. To assist a new admin the admin help tooltip was created for account management. Template and conference management have not yet been implemented.



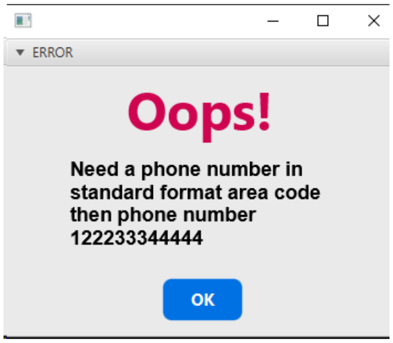
Oops!

Must contain an email address in username



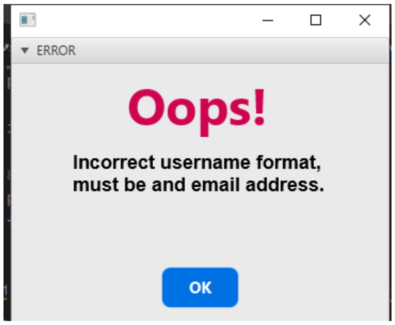
Oops!

Cannot delete your admin own account.



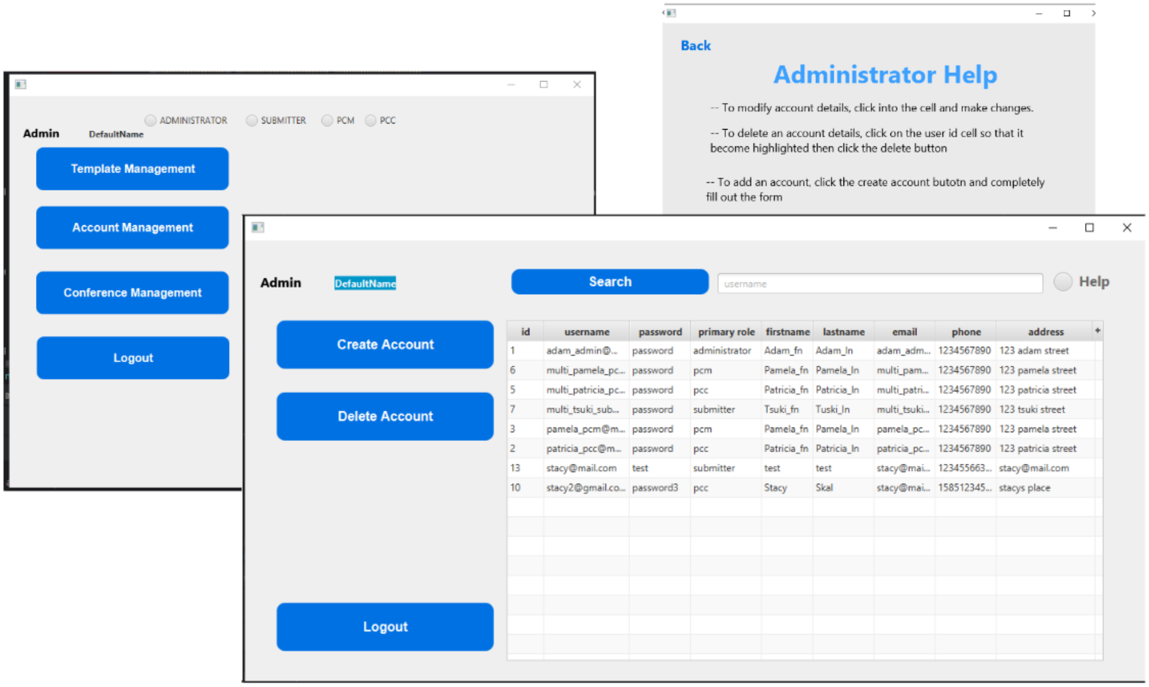
Oops!

Need a phone number in standard format area code then phone number 122233344444



Oops!

Incorrect username format, must be and email address.



Administrator Help

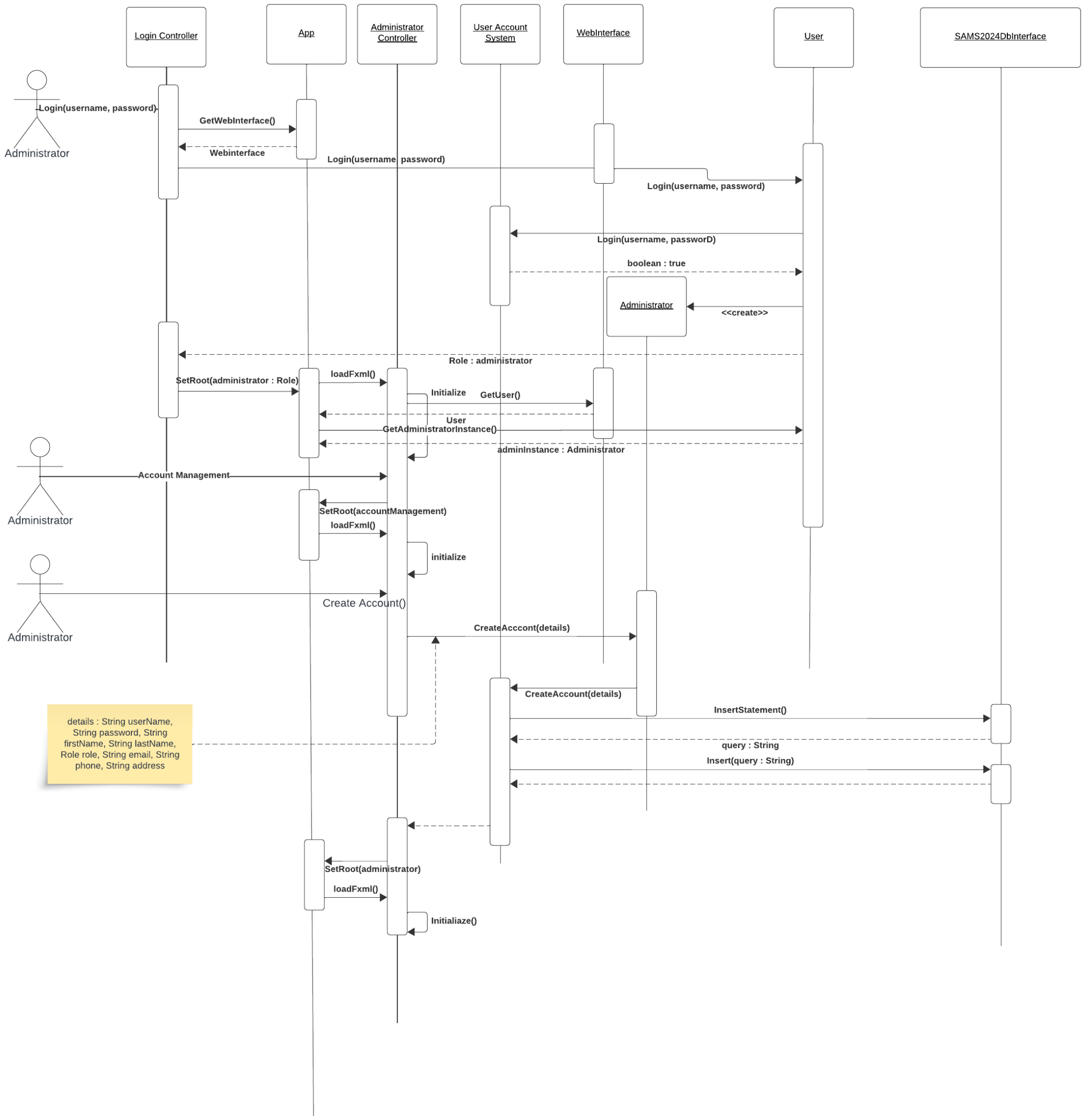
-- To modify account details, click into the cell and make changes.

-- To delete an account details, click on the user id cell so that it become highlighted then click the delete button

-- To add an account, click the create account butotn and completely fill out the form

id	username	password	primary role	firstname	lastname	email	phone	address
1	adam_admin@...	password	administrator	Adam_fn	Adam_ln	adam_adm...	1234567890	123 adam street
6	multi_pamela_pc...	password	pcm	Pamela_fn	Pamela_ln	multi_pam...	1234567890	123 pamela street
5	multi_patricia_pc...	password	pcc	Patricia_fn	Patricia_ln	multi_patri...	1234567890	123 patricia street
7	multi_tsuki_sub...	password	submitter	Tsuki_fn	Tuski_ln	multi_tsuki...	1234567890	123 tsuki street
3	pamela_pcm@m...	password	pcm	Pamela_fn	Pamela_ln	pamela_pc...	1234567890	123 pamela street
2	patricia_pcc@m...	password	pcc	Patricia_fn	Patricia_ln	patricia_pc...	1234567890	123 patricia street
13	stacy@mail.com	test	submitter	test	test	stacy@mai...	123455663...	stacy@mail.com
10	stacy2@gmail.co...	password3	pcc	Stacy	Skal	stacy@mai...	158512345...	stacys place

- Sequence Diagram for the main flow of the administrator creating an account.
    - This administrator account management flow is shown as it was implemented in the source code. The administrator logs in from the login controller. The login controller utilized the instance of



Tools :

We employed a diverse range of tools and technologies to create an efficient and user-friendly application. We started with MySQL Workbench for our database needs, a choice driven by its robustness and flexibility in managing complex data structures. Using MySQL Workbench we designed our database tables and established intricate relationships using foreign keys, ensuring a seamless flow of data across the system. When it came to the user interface design, we initially considered React for its modular architecture and seamless API integration, aiming to develop an interactive web-based client-side. However, faced with a tight deadline of 3-4 weeks, we adapted our strategy and shifted to JavaFX for GUI creation. This strategic shift enabled us to rapidly construct a visually appealing and functional desktop interface, utilizing wireframes for effective layout planning. Parallelly, in the realm of server-side development, we chose the Java programming language for its robust and secure features, complemented by the use of IntelliJ IDEA as our IDE. This combination was instrumental in developing a back-end architecture that was stable and efficient, crucial for the seamless operation of our application's visual and interaction design."

Appendix

Citations/Implementation References:

- Ahmed NabilAhmed Nabil 17.8k1212 gold badges6363 silver badges8888 bronze badges, et al. “How to Initialize List Object in Java?” *Stack Overflow*, 1 Jan. 1959, stackoverflow.com/questions/13395114/how-to-initialize-liststring-object-in-java.
- “Arrays in Java.” *GeeksforGeeks*, GeeksforGeeks, 10 Nov. 2023, www.geeksforgeeks.org/arrays-in-java/.
- “Connectors and Apis Manual :: 3.3.2 Installing Connector/J Using Maven.” *MySQL*, dev.mysql.com/doc/connectors/en/connector-j-installing-maven.html. Accessed 3 Dec. 2023.
- “Database Tool Window: Intellij IDEA.” *IntelliJ IDEA Help*, www.jetbrains.com/help/idea/database-tool-window.html#view_options. Accessed 3 Dec. 2023.
- “Database Tools and SQL: Intellij IDEA.” *IntelliJ IDEA Help*, www.jetbrains.com/help/idea/relational-databases.html. Accessed 3 Dec. 2023.
- Dori, et al. “Best Way to Create Enum of Strings?” *Stack Overflow*, 1 Dec. 1956, stackoverflow.com/questions/3978654/best-way-to-create-enum-of-strings.
- “Enum Types.” *Enum Types (The JavaTM Tutorials > Learning the Java Language > Classes and Objects)*, docs.oracle.com/javase/tutorial/java/javaOO/enum.html#:~:text=An%20enum%20type%20is%20a,the%20days%20of%20the%20week. Accessed 3 Dec. 2023.
- *How to Add Javafx to an Intellij Project*, javabook.bloomu.edu/setupjavafx.html. Accessed 3 Dec. 2023.
- “How to Easily Insert Datetime in Mysql?” *Online Tutorials, Courses, and eBooks Library*, www.tutorialspoint.com/how-to-easily-insert-datetime-in-mysql. Accessed 3 Dec. 2023.
- HristoHristo 45.8k6565 gold badges164164 silver badges230230 bronze badges, et al. “What Is the Default Initialization of an Array in Java?” *Stack Overflow*, 1 Sept. 1956, stackoverflow.com/questions/3426843/what-is-the-default-initialization-of-an-array-in-java.
- *Intellij Not Able to Resolve Classes of External Libraries for Maven ...*, intellij-support.jetbrains.com/hc/en-us/community/posts/207108385-IntelliJ-not-able-to-resolve-classes-of-external-Libraries-for-Maven-projects. Accessed 3 Dec. 2023.
- “Java Exceptions - Try...Catch.” *Java Exceptions (Try...Catch)*, www.w3schools.com/java/java_try_catch.asp. Accessed 3 Dec. 2023.
- Jenkov, Jakob. “JavaFX ComboBox.” *Jenkov.Com Tech & Media Labs - Resources for Developers, IT Architects and Technopreneurs*, jenkov.com/tutorials/javafx/combobox.html#:~:text=The%20JavaFX%20ComboBox%20control%20enables,represented%20by%20the%20class%20javafx. Accessed 3 Dec. 2023.
- “Maven Dependencies: Intellij Idea.” *IntelliJ IDEA Help*, www.jetbrains.com/help/idea/work-with-maven-dependencies.html#generate_maven_dependency. Accessed 3 Dec. 2023.

- “MySQL - Insert Current Date/Time?” *Online Tutorials, Courses, and eBooks Library*, www.tutorialspoint.com/mysql-insert-current-date-time#:~:text=To%20insert%20only%20date%20value,can%20use%20now()%20method.&text=Insert%20both%20date%20and%20time%20with%20the%20help%20of%20now(). Accessed 3 Dec. 2023.
- “MySQL 8.0 Reference Manual :: 3.3.4.3 Selecting Particular Columns.” *MySQL*, dev.mysql.com/doc/refman/8.0/en/selecting-columns.html. Accessed 3 Dec. 2023.
- “MySQL Workbench 8.0.34.” *MySQL*, dev.mysql.com/downloads/workbench/. Accessed 3 Dec. 2023.
- “MySQL: Intellij Idea.” *IntelliJ IDEA Help*, www.jetbrains.com/help/idea/mysql.html#2459fcec. Accessed 3 Dec. 2023.
- RAJMOHANRAJMOHAN 51511 gold badge55 silver badges1414 bronze badges, et al. “How to Connect Intellij with Local Mysql?” *Stack Overflow*, 1 Sept. 1962, stackoverflow.com/questions/38660104/how-to-connect-intellij-with-local-mysql.
- Ravichandra NamburiRavichandra Namburi 10388 bronze badges, and Manuel PerezManuel Perez 3711 gold badge44 silver badges55 bronze badges. “How to Add Button to TableView Using FXML?” *Stack Overflow*, 1 July 1963, stackoverflow.com/questions/44005787/how-to-add-button-to-tableview-using-fxml.
- “Release: Javafx 2.2.40.” *Getting Started with JavaFX: Using FXML to Create a User Interface | JavaFX 2 Tutorials and Documentation*, 30 Aug. 2013, docs.oracle.com/javafx/2/get_started/fxml_tutorial.htm.
- “Release: Javafx 2.2.” *Using JavaFX UI Controls: Table View | JavaFX 2 Tutorials and Documentation*, 27 Aug. 2013, docs.oracle.com/javafx/2/ui_controls/table-view.htm.
- “Run Queries: Intellij Idea.” *IntelliJ IDEA Help*, www.jetbrains.com/help/idea/run-a-query.html#run_statements_in_a_query_console. Accessed 3 Dec. 2023.
- SHILPA ARSHILPA AR 33222 gold badges77 silver badges1818 bronze badges, et al. “Which Is the Best Way to Append Single Quotes for a String in Java.” *Stack Overflow*, 1 May 1961, stackoverflow.com/questions/29516260/which-is-the-best-way-to-append-single-quotes-for-a-string-in-java.
- *SQL DELETE STATEMENT*, www.w3schools.com/sql/sql_delete.asp. Accessed 3 Dec. 2023.
- *SQL Drop Database Statement*, www.w3schools.com/sql/sql_drop_db.asp. Accessed 3 Dec. 2023.
- *SQL Drop Table Statement*, www.w3schools.com/sql/sql_drop_table.asp. Accessed 3 Dec. 2023.
- *SQL NOT OPERATOR*, www.w3schools.com/sql/sql_not.asp. Accessed 3 Dec. 2023.
- Trettevik, Joseph. “Git and Github: How to Revert a Single File.” *DEV Community*, DEV Community, 31 Aug. 2020, dev.to/lofiandcode/git-and-github-how-to-revert-a-single-file-dha.
- user_vsuser_vs 1, et al. “Produce a Comma Separated List Using Stringbuilder.” *Stack Overflow*, 1 Dec. 1961, stackoverflow.com/questions/32987478/produce-a-comma-separated-list-using-stringbuilder. '
- Panchal Aayush and Skalicky, Stacy and Chander, Poorna and Puttaparthi, Parinay SWEN-732/Scrum for A Day : Fxml and Java Controller Class Source Code.

- https://rfic-ieee.org/sites/rfic/files/content_images/files/RFIC_Paper_WORD_Template_Letter.docx