**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering) Shahbad Daulatpur, Bawana Road, Delhi-110042

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CS209n: Software Engineering**

**Project Report**

**Submitted By:**

Divyam Goyal 23/CS/145

Arihant Jain 23/CS/075

Semester-3

**Submitted to:**

Prof. Rohit Beniwal

Department of Computer

Science and Engineering

**ACKNOWLEDGEMENT**

We would like to express our heartfelt gratitude to **Mr. Rohit Beniwal** for his invaluable guidance and continuous support throughout the development of this Software Requirements Specification (SRS) for the **Conference Management System**. His expertise and insights have been instrumental in shaping this project, and we are truly grateful for the opportunity to learn under his mentorship.

We also extend our appreciation to everyone who contributed directly or indirectly to this project by providing their feedback and support. Their encouragement has been a source of motivation during the completion of this document.

This work is the result of the collaborative efforts of **Arihant Jain(23/CS/075)** and **Divyam Goyal(23/CS/145)**, whose dedication and teamwork made this project possible.

Thank you for your guidance and support.

**Arihant Jain (23/CS/075)  
Divyam Goyal (23/CS/145)**

**DECLARATION**

We hereby declare that the Software Requirements Specification (SRS) document for the **Conference Management System (CMS)** has been prepared by us, **Arihant Jain(23/CS/075)** and **Divyam Goyal(23/CS/145)**, as part of our academic endeavour.

This document outlines the functional, non-functional, hardware, software, and performance requirements for the development and implementation of the Conference Management system. The content of this document has been thoroughly analysed and is intended to serve as a comprehensive guide for all stakeholders involved in the project.

We affirm that the work presented in this SRS document is original and has been developed with due diligence to meet the intended objectives of the Conference Management System.

**ABSTRACT**

The **Conference Management System (CMS)** is a comprehensive platform designed to streamline the organization and management of academic and professional conferences. This system automates critical processes such as participant registration, paper submission, review and evaluation, scheduling, and communication between stakeholders. It provides a centralized solution that enhances efficiency, reduces manual effort, and ensures a smooth workflow for conference organizers, authors, and reviewers.

Key features of the CMS include user-friendly interfaces for easy navigation, role-based access control for different stakeholders, plagiarism detection for paper submissions, real-time notifications, and integrated tools for feedback and decision-making. The system is built with scalability in mind, supporting both physical and virtual conference setups.

**Table of Contents**

1. Introduction  
   1.1. Purpose

1.2. Scope  
1.3. Definitions, Acronyms and Abbreviations  
1.4. References   
1.5. Overview

2. Overall Description

2.1. Product Perspective

2.2. Product Functions

2.3. User Characteristics

2.4. Operating Environment

2.5. Design and Implementation Constraints

3. System Features

3.1. Registration

3.2. Login

3.3. Submit Paper

3.4. Check Status/Feedback

3.5. Manage Conference

3.6. Assign Reviewers

3.7. Submit Revised Paper

3.8. Review a Paper

3.9. Submit Feedback

3.10. Logout

4. External Interface Requirements

4.1. User Interface

4.2. Hardware Interfaces

4.3. Software Interfaces

4.4. Communication Interfaces

5. Other Non Functional Requirements

5.1. Performance Requirements

5.2. Safety Requirements

5.3. Security Requirements

5.4. Software Quality Attributes

6. Use Case Template

7. Diagrams

8. Testing

1. **INTRODUCTION**

This Document defines the overall software requirements for the ‘Conference Management System.’ Efforts have been made to determine the requirements accurately. The final product will have only features/ functionalities mentioned in this document, and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing / testing / implementing this product. If it requires some extra features, a formal change request will need to be made, and subsequently, a new release of this document and/ or product will be produced.

**1.1 PURPOSE**

This specification document describes the capabilities that will be provided by the software application ‘Conference Management System’. It also states the various required constraints by which the system will abide. The intended audience for this document is the development team, testing team, and end users of the product.

**1.2 SCOPE**

The scope of the Conference Management System (CMS) revolves around managing the submission, review, and decision-making processes for academic and professional conferences. Designed to support the roles of Authors, Reviewers, and Chairs, the system facilitates the end-to-end workflow of conference organization. It is tailored to meet the needs of academic communities by providing a robust, secure, and efficient platform.

The CMS will allow Authors to submit their research papers or abstracts to specific conference tracks or topics. It will support functionalities such as file upload, version management, and the ability to track submission statuses. Authors will also have the option to revise their submissions based on reviewer feedback before final decisions are made.

**1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS**

The following abbreviations/acronyms have been used throughout this document:

1. **CMS:** Conference Managment System

**1.4 REFERENCES**

i. EasyChair - [***https://easychair.org/***](https://easychair.org/)

ii. Microsoft CMT - [***https://cmtint.research.microsoft.com/***](https://cmtint.research.microsoft.com/)

iii. IEEE Recommended Practice of Software Specification- IEEE std 830-1993.

**1.5 OVERVIEW**

The rest of this SRS document describes the various system requirements interface, features and functionalities in detail.

1. **OVERALL DESCRIPTION**

Conference Management System is a web-based software product catering to the arrangement and coordination of academic and professional conferences on a large scale (like EasyChair or Microsoft CMT). The system is dedicated to providing end-to-end solutions for the management of the complete lifecycle of a conference, particularly paper submission, review processes, scheduling, and participant management.

This application will allow conference organizers to set up details of their events, such as deadlines, tracks, and themes. It allows participants (authors) to submit a paper, abstract, or proposal directly in the system and reviewers to evaluate submissions efficiently.

Key features include user role management to define and control access for chairs, authors, reviewers, and program committee members. Organizers can create conference tracks or sessions, assign reviewers to papers, send notifications, and finalize acceptance or rejection of submissions.

**2.1 PRODUCT PERSPECTIVE**

The CMS is a web application that operates on the server-side and utilizes an external relational database management system for its data layer. Users can access the system through any web browser without additional software. The system will be built using MySQL technologies, ensuring efficient and fast development.

**2.2 PRODUCT FUNCTIONS**

The Conference Management System will provide the following features:

* Registration
* Login
* Submit Paper
* Check Status/ Feedback
* Manage Conferences
* Assign Reviewers
* Submit Revised Paper
* Review a Paper
* Submit Feedback
* Logout

**2.3. USER CHARACTERISTICS**

CMS is mainly meant to be used by academic staff. However, the system will not require any higher technical knowledge and experience with similar applications. Thus it can be used by general users, too.

**2.4. OPERATING ENVIRONMENT**

CMS can run on every platform where a Web Browser exists. The web server should be reliable. Server uptime is critical around author and reviewer deadlines. Web servers and DBMS should be capable of receiving some requests simultaneously. The system shall possess these features on a server.

**2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS**

To ensure maintainability, the system should be built using widely accepted frameworks and technologies. For instance:

* Backend: Node js, Express js.
* Frontend: HTML, CSS, BootStrap, React.js.
* Database: MySQL or MongoDB.

**3. SYSTEM FEATURES**

**3.1. REGISTRATION**

*3.1.1. Description*

In this feature, the User (Author, Reviewer, Chair) information is put into the database. Once registered, the user can log in to the CMS and access its other functionalities.

**Providing Personal Data**: In this step, users will have to fill in their personal data. Personal Data includes:

* First Name\*
* Middle Name
* Last Name\*
* Qualification\*
* Organization/University Name\*
* Country/ Region Name\*
* Email Address\*
* Google Scholar ID
* Password\*

Fields marked with an asterisk (\*) are mandatory when completing the registration form.

*3.1.2 Functional Requirements*

* Passwords are required to contain 12 characters, each of which should be lowercase letters, uppercase letters, numbers, and symbols. Valid symbols are: ~`!@#$%^&\*()-\_=+{}[]|:;"'<>,.?/
* The server should reject any email that has already been registered.
* The Server should check for the validity of Google Scholar ID.

**3.2. LOGIN**

*3.2.1 Description*

The login process ensures secure access to the system and requires user credentials. The authentication process includes the following components:

1. E-mail Field
   1. Users must enter their registered e-mail addresses in this field.
   2. The e-mail address uniquely identifies each user within the system.
2. Password Field
   1. Users must enter their passwords to access the system.
   2. After successful login, users will have the option to change their passwords to maintain account security.

This Single Login Authentication mechanism helps safeguard user information and ensures only authorized access to the system.

*3.2.2. Functional Requirements*

1. The server must verify the validity of the user's entered e-mail and password.

**3.3. SUBMIT PAPER**

*3.3.1 Description*

Authors invited to a conference will be able to submit their papers.

The paper submission process includes the following:

1. **Paper Submission Page**: A dedicated page for authors to upload their papers for the conference.
2. **Paper Format Validation**: Before submission, authors must ensure their paper meets the required format specifications. The paper should be in PDF format and comply with certain constraints, such as:
   1. Margins.
   2. Paper size (e.g., Letter or A4).
   3. File size, which must not exceed a specified limit.

*3.3.2 Functional Requirements:*

1. User Authentication
   1. The system must allow only registered and authorized authors to access the paper submission page.
2. Paper Upload
   1. Authors must be able to upload their papers in PDF format via the submission page.
3. Plagiarism Check
   1. The system must perform an automated plagiarism check on the uploaded paper.
   2. Authors should receive a report indicating the plagiarism percentage. Papers exceeding the allowable plagiarism threshold will be rejected or flagged for revision.
4. File Format Validation
   1. The system must ensure that the uploaded file is in PDF format.
   2. Non-PDF files should be rejected with an appropriate error message.
5. File Size Constraints
   1. The system must enforce a maximum file size limit (e.g., 10 MB).
   2. Files exceeding the size limit should be rejected, and users should be notified.

**3.4. CHECK STATUS/ FEEDBACK**

*3.4.1. Description*

The "Check Status/Feedback" functionality in the Conference Management System enables authors to monitor the status of their submitted papers and access feedback provided by reviewers or conference organizers. This feature promotes transparency in the review process and supports authors in making necessary revisions.

### *3.4.2. Functional Requirements*

1. Login Authentication
   1. Only registered users can access the "Check Status/Feedback" feature after logging into their accounts.
2. Paper Submission Status
   1. The system must display the current status of each paper submitted by the author, including:
      1. Under Review
      2. Accepted
      3. Rejected
      4. Revision Required
3. Reviewer Feedback
   1. The system must provide comprehensive feedback for each paper submitted.
   2. Feedback should include:
      1. Reviewer comments and recommendations for improvement.
      2. Ratings or scores, if applicable.
   3. Access to feedback must be restricted to the author(s) of the respective paper.
4. Notification of Status Updates
   1. Authors should be notified via e-mail or within the system whenever their paper's status is updated or feedback is available.
5. Revision History Tracking
   1. The system must allow authors to view a detailed history of submitted revisions along with the corresponding feedback.
6. Downloadable Feedback Reports
   1. Authors must have the option to download detailed feedback reports for offline reference.

**3.5. MANAGE CONFERENCE**

*3.5.1. Description*

The "Manage Conference" feature in a Conference Management System (CMS) enables organizers to oversee and manage all aspects of a conference. This includes creating and editing conference details, managing schedules, monitoring participant registrations, handling submissions, assigning reviewers, and communicating with attendees. The feature ensures that organizers have full control over the conference workflow to ensure its successful execution.

*3.5.2. Functional Requirements*

1. Create Conference
   1. The CMS must allow organizers to create a new conference by providing details like Name, Description, time, submission details etc.
2. Edit Conference Details
   1. The CMS should have a feature of edit or update conference details including deadlines schedules as required.
   2. The CMS should provide functionality to create, modify, and publish the conference schedule, including session details, speakers, and timings.
3. Researchers Registration Management
   1. The CMS should enable organizers to track researchers registration and export registration details.
4. Paper Submission and Review Management
   1. The CMS must allow organizers to monitor the status of submitted papers.
   2. Organizers should be able to assign papers to reviewers and track the progress of reviews.
5. Notifications and Announcements
   1. The CMS must enable organizers to send notifications or announcements to authors, reviewers, and authors about important updates or reminders.

**3.6. ASSIGN REVIEWERS**

*3.6.1 Description*

The "Assign Reviewers" feature in a Conference Management System enables program chairs to allocate submitted papers to suitable reviewers based on their expertise. The system simplifies reviewer management, balances workloads, and ensures papers are reviewed by qualified experts.

### *3.6.2 Functional Requirements*

1. Automated Reviewer Assignment
   1. The system must support automated assignment of reviewers to papers based on:
      1. Matching keywords from the paper and reviewer profiles
      2. Reviewer workload or availability
      3. Conflict-of-interest checks
   2. Organizers should be able to adjust assignments manually if needed.
2. Conflict of Interest Management
   1. The system must detect and prevent conflicts of interest by:
      1. Allowing reviewers to declare conflicts for specific authors or papers
      2. Automatically checking for affiliations or prior collaborations between reviewers and authors
3. Reviewer Load Balancing
   1. The system must distribute papers evenly among reviewers, ensuring no reviewer is overloaded.
   2. Chairs must have the ability to override load limits when necessary.
4. Reviewer Notification
   1. The system must notify reviewers when papers are assigned to them, including:
      1. Paper title and abstract
      2. Submission deadlines
      3. Instructions for completing the review
5. Tracking and Monitoring Reviewer Progress
   1. Chairs must be able to monitor the review progress of each reviewer, including:
      1. Number of assigned papers
      2. Status of completed reviews
      3. Review deadlines
6. Reassignment of Papers
   1. The system must allow Chairs to reassign papers in cases where a reviewer is unable to complete the review.
7. Reviewer Feedback Management
   1. The system must ensure reviewers can provide their feedback and scores within the platform.
   2. Chairs must be able to view and manage reviewer feedback as part of the evaluation process.

**3.7. SUBMIT REVISED PAPER**

*3.7.1. Description*

The "Submit Revised Paper" feature in Conference Management System allows authors to upload updated versions of their papers in response to reviewer feedback.

### *3.7.2. Functional Requirements*

1. Access to Revision Submission
   1. Authors must log into the system to access the revision submission functionality.
   2. Only authors of papers flagged as "Revision Required" are permitted to submit revised versions.
2. Feedback Access
   1. Authors must be able to view feedback provided by reviewers, which includes:
      1. Comments and recommendations for improvement.
      2. Specific issues to be addressed in the revised paper.
3. Uploading Revised Papers
   1. The system must enable authors to upload their revised papers.
   2. The revised submissions must adhere to predefined requirements, including:
      1. Acceptable file format (e.g., PDF).
      2. Maximum allowable file size.
      3. Compliance with formatting rules (e.g., margins, font style, and page limits).
4. Adherence to Deadlines
   1. Authors are required to submit their revised papers before the specified deadline.
   2. The system must send reminders of approaching deadlines via email or system notifications.
5. Plagiarism Detection
   1. The system must conduct a plagiarism check on the uploaded revised paper.
6. Communication with Reviewers
   1. Authors should be able to include a "Response to Reviewers" document summarizing the changes made based on the feedback.
   2. Reviewers and chairs must have access to this document when evaluating the revised submission.
7. Notification System
   1. Authors must receive notifications for:
      1. Successful submission of their revised paper.
      2. Any further required actions or updates related to the revision.

**3.8. REVIEW A PAPER**

*3.8.1. Description*

The "Review a Paper" feature in a Conference Management System allows assigned reviewers to evaluate and provide detailed feedback on submitted papers. This functionality simplifies the peer-review process, ensuring that submissions are assessed based on established criteria, such as originality, relevance, methodology, and clarity. It promotes efficient communication between reviewers and organizers while maintaining confidentiality and transparency, especially in blind review settings.

### *3.8.2. Functional Requirements*

1. Access to Assigned Papers
   1. Reviewers must log into the system to view papers assigned to them for review.
   2. Only papers explicitly allocated to a reviewer should be accessible.
2. Confidentiality and Anonymity
   1. The system must implement anonymity based on the chosen review type:
      1. Double-blind review: Both the authors' and reviewers' identities remain concealed.
      2. Single-blind review: Authors' identities are hidden, but reviewers' identities are visible to organizers.
   2. Author details must be restricted unless the selected review type allows access.
3. Display of Review Criteria
   1. The system must present evaluation criteria, including:
      1. Originality and innovation.
      2. Relevance to the conference theme.
      3. Clarity, coherence, and structure.
4. Providing Feedback and Scores
   1. Reviewers must be able to provide detailed feedback, including:
      1. Written comments highlighting strengths, weaknesses, and areas for improvement.
      2. Scores or ratings for specific aspects, such as originality and overall quality.
5. Recommendation for Decision
   1. Reviewers must submit a decision for each paper, choosing from options such as:
      1. Accept.
      2. Accept with minor revisions.
      3. Major revisions required.
      4. Reject.
6. Draft and Edit Reviews
   1. Reviewers must have the ability to save a draft of their review and complete or edit it later.
   2. Edits must be allowed until the review submission deadline.
7. Conflict of Interest Management
   1. Reviewers must declare any conflicts of interest before accessing a paper.
   2. If a conflict is reported, the system must allow program chairs to reassign the paper.
8. Review Deadlines
   1. The system must enforce deadlines for review submissions by sending notifications about upcoming due dates via email or system alerts.
9. Monitoring and Reporting
   1. Organizers must be able to track the status of reviews, including:
      1. Progress of reviews for each assigned paper.
      2. Reviewer performance metrics (e.g., punctuality and quality of feedback).
   2. The system must generate summary reports on review outcomes.

**3.9. SUBMIT FEEDBACK**

*3.9.1. Description*

The "Submit Feedback" feature in a Conference Management System (CMS) allows reviewers to provide detailed evaluations of submitted papers. Reviewers assess papers based on predefined criteria such as originality, relevance, clarity, and technical quality. This feature ensures that feedback is structured, comprehensive, and delivered efficiently to authors, helping them refine their work. The process also aids conference organizers in determining which submissions meet the standards for acceptance.

*3.9.2. Functional Requirements*

1. Provide Structured Feedback
   1. Reviewers must be able to provide feedback through a structured form with fields for specific evaluation criteria, such as Originality, Relevance to the Conference theme, Clarity of Content, technical quality etc.
2. Upload Additional Feedback Documents
   1. Reviewers must have the option to upload supplementary feedback files, such as annotated versions of the paper or detailed review reports.

**3.10. LOGOUT**

*3.10.1. Description*

The "Logout" functionality in a Conference Management System enables users to safely exit the platform once their tasks are completed. It ensures that their session is fully terminated, safeguarding sensitive data from unauthorized access. This feature is essential for ensuring privacy and security, particularly when users access the system on shared or public devices. After logging out, users are redirected to the login page, requiring them to re-authenticate in order to regain access to the system.

*3.10.2. Functional Requirements*

1. Access Control

a. Only authenticated users (i.e., users who have logged into the system) can log out.

b. The logout option must be available to all users with an active session, including authors, reviewers, and organizers.

1. Session Termination

a. The system must terminate the user session upon logging out, ensuring that no further actions can be performed without re-authentication.

b. All session-related data (e.g., temporary files, session tokens) must be securely cleared from the system.

1. Redirect to Login Page

a. After logout, users must be redirected to the login page to prevent unauthorized access to any system functionality.

1. Confirmation of Logout

a. Users should receive a confirmation message or notification that they have successfully logged out.

b. If a user attempts to navigate to any protected area after logging out, the system must redirect them back to the login page.

1. Automatic Logout

a.The system may include an automatic logout feature that logs users out after a specified period of inactivity (e.g., 30 minutes).

b. Users should be notified before automatic logout occurs allowing them to extend their session.

**4. EXTERNAL INTERFACE REQUIREMENTS**

**4.1. USER INTERFACES**

These requirements aims to ensure a seamless and efficient user experience for all participants in the conference management system.

1. The User Interface must provide a user-friendly navigation menu with options like:
   1. Dashboard
   2. Paper Submission
   3. Review Process
   4. Conference Schedule
   5. About us
   6. Login/Logout
   7. Notifications
2. The CMS must provide a registration and login interface.
3. The CMS must provide customized dashboards for different user roles (e.g., Author, Reviewer, Chair) with role-specific features and access.
4. The CMS must provide a paper submission page with fields to upload files, enter paper details (e.g., title, abstract, keywords), and validate formatting requirements.
5. The CMS must provide a responsive user interface that adapts across various devices, including desktops, laptops, tablets, and smartphones.
6. The CMS must offer a clear, interactive schedule interface displaying session details, paper presentations, speaker information, and time slots.
7. The CMS must include a notifications panel to inform users about deadlines, paper status updates, and conference announcements.
8. The CMS must provide a reviewer interface to access submitted papers, provide feedback, and assign ratings.

**4.2. HARDWARE INTERFACES**

1. A minimum screen resolution of 800 x 600 is required to ensure proper and complete display of screens. Higher resolutions are fully supported without any issues.
2. Dedicated hardware for Research files backup, such as a cloud-based backup solutions, must be supported to ensure data integrity.
3. Software should also support Scaling which manages load during peak usage such as near submission deadlines.
4. The CMS must be hosted on the server with the following minimum specifications.
   1. Processor should be multicore server grade CPU.
   2. It should have at least memory of 16GB RAM.
   3. For storage, 1TB or more (preferably SSD for faster performance).
5. The CMS should function over a reliable internet connection, with the following minimum requirements:
   1. A user should be able to use the Conference Management System (CMS) with an internet connection speed of 2 Mbps or higher.

**4.3. SOFTWARE INTERFACES**

1. The CMS should be accessible through systems running modern operating systems, including Windows (10,11), macOS, Linux,  
   and Mobile devices.
2. The CMS must be compatible with the latest versions of modern web browsers, such as Google Chrome, Microsoft Edge, Brave, Mozilla Firefox, Safari, etc. The browser must support JavaScript, HTML5, and CSS3.
3. The CMS should integrate with third-party plagiarism detection software (e.g., Turnitin or Copyscape) to verify the originality of uploaded papers.
4. The CMS should provide APIs for integration with third party services such as Payment Gateway for Conference Fees.
5. The CMS must include a built-in PDF viewer to allow users to view submitted papers without requiring external software.

**4.4. COMMUNICATION INTERFACES**

None

**5. OTHER NON FUNCTIONAL REQUIREMENTS**

**5.1. PERFORMANCE REQUIREMENTS**

The system's performance is determined by its response time and the speed of data submission. As a real-time application, the system requires quick response times, which rely on the efficiency of the implemented algorithms. In its initial version, the system will have limited file submission speeds, reducing the need for high network capacity. However, as usage increases, network requirements and file submission speed may scale accordingly.

**5.2 SAFETY REQUIREMENTS**

The system must verify the following:

1. The HTML content is syntactically well-formed.
2. Web forms are consistent with the services processing their input.
3. Hyper-links maintain referential integrity in both static and dynamically generated content.
4. The code for session operations is statically bound to variables defined within the session scope.

In the event of an error, the system should provide users with clear and appropriate help messages.

**5.3. SECURITY REQUIREMENTS**

To ensure system security, a database replication technique should be employed to safeguard all critical data. In the event of a crash, the system must support data backup and recovery to prevent loss.

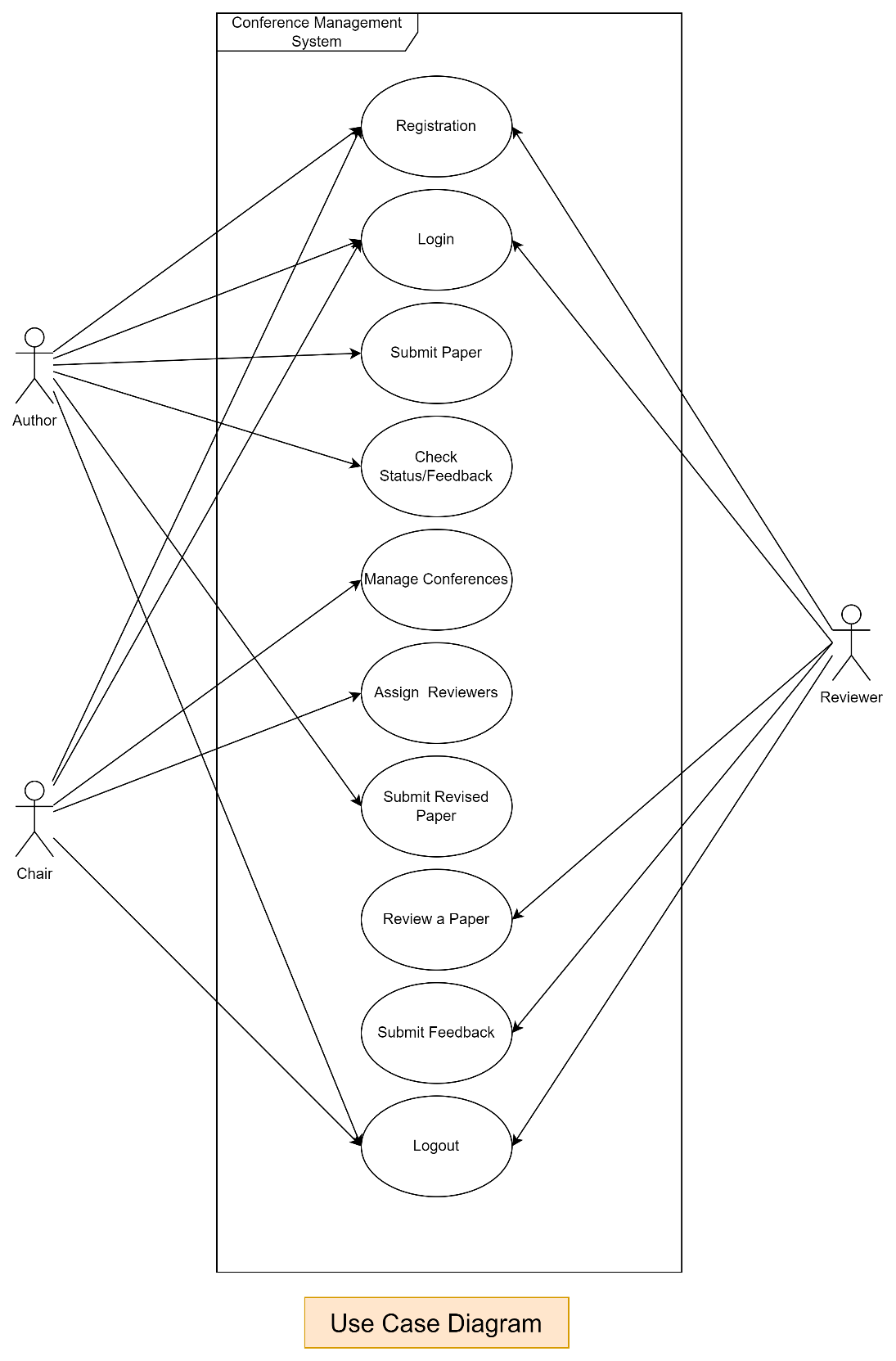
**5.4. SOFTWARE QUALITY ATTRIBUTES**

The system will feature a straightforward and user-friendly graphical interface, enabling users to easily understand and access all website functionalities. Actions can be performed quickly and efficiently with minimal clicks.

**6. USE CASE TEMPLATE**

**Stakeholders:**

* **Author -** Interested in submitting a paper, by completing a form asking for details about it, in order to obtain the review of qualified people. After the review period has passed .if it is positive, the paper will be presented to the public .
* **Program chair:** Interested in having the papers review by qualified peoples.
* **Reviewer:** Interested in analyzing and review new ideas.



**Use Case 1**

**Use case name** : Registration

**1. Introduction**

In this feature, the User (Author, Reviewer, Chair) information is put into the database. Once registered, the user can log in to the CMS and access its other functionalities

**2. Actor**

**2.1. Primary actor** : Author , Reviewer, Chair

**2.2. Secondary Actor**: None

**3. Preconditions**

* A Registration scenario is opened through a Registration operation

**4. Postconditions**

* The account information is saved in the application database
* Project database should not suffer any change if only save operations were executed with success .

**5. Basic Flow**

1. This use case starts when the actor wants to create an account
2. User proceeds to registration page
3. User selects its role(Chair ,Author,Reviewer)
4. The user enters the Personal Information (Name, age, phone no. ,etc.)
5. The User enters the Academic Information (Educational qualification ,year of graduation, specialisation)
6. User enters its username and password .
7. The system will indicate that the information has been registered to the system
8. The user exits the system

**6. Alternative Flows**

This alternative flow applies when the passwords do not match

This flow can occur after step 6

6.1 The author re-enters the information

6.2 The system will validate the information

6.3 The use case continues with step

This alternative flow applies if the username already exists

This flow can occur after step 3

6.1 The author re-enters the information

6.2 The system will validate the information

6.3 The use case continues with step

**7. Special Requirements**

None

**Use Case 2**

**Use case name** : Login

**1. Introduction**

The login process ensures secure access to the system and requires user credentials.

**2. Actor**

**2.1. Primary actor** :Chair, Author , Reviewer

**2.2. Secondary Actor**: None

**3. Preconditions**

* A login scenario is opened through a Login operation
* User must be registered

**4. Postconditions**

* User will be able to access all the features that are entitled to it .

**5. Basic Flow**

1. User proceeds to login page
2. The user enters the identification data for the account (e.g. Username, Password)
3. The system checks the data
4. The user continues with its next step

**6. Alternative Flows**

This alternative flow applies when the password is not correct

This flow can occur after step 3

1. The author re-enters the information
2. The system will validate the information
3. The use case continues with step

This alternative flow applies if the username is not correct

This flow can occur after step 3

1. . The author re-enters the information
2. . The system will validate the information
3. . The use case continues with step

**7. Special Requirements**

None

**Use Case 3**

**Use case name** : Submit Paper

**1. Introduction**

Authors invited to a conference will be able to submit their papers.

**2. Actor**

**2.1. Primary actor** : Author

**2.2. Secondary Actor**: None

**3. Preconditions**

* The author must be logged in and be authorized to perform this use case
* A submission scenario is opened through an Add Submission operation
* The state of the Conference must be “Submissions are opened” in order to perform a submission. Paper cannot be submitted if the conference is in “Submission are not started” or “Review period started” state.

**4. Postconditions**

* The submission information is saved in the application database
* Conference database should remain in a consistent state if the Save/Commit operation has ended with success.
* Conference entities should not suffer any changes if the Save/Commit operation has not ended with success.
* Reasearch Paper database should not suffer any change if only save (not commit) operations were executed with success on scenario

**5. Basic Flow**

This use case starts when the actor wants to create/modify an entry for the conference

1. Author proceeds to the conference submission.
2. The system checks the conference timeline .
3. The author proceeds to the page that contains the submission form .
4. The author enters the credentials for the submission (e.g. Name, Keywords) .
5. The author selects the physical file he/she wants to upload as paper .
6. The system will indicate that the submission has been registered to the system .
7. The user checks the status of its submission .
8. The user exits the system .

**6. Alternative Flows**

This alternative flow applies when the submissions are not opened yet

This flow can occur after step 2

2.1. The use case ends

This alternative flow applies if the paper is not uploaded

This flow can occur after step 5

5.1 The author re-enters the information

5.2 The system will validate the information

5.4 The use case continues with step

**7. Special Requirements**

None

**Use Case 4**

**Use case name** : Review paper

**1. Introduction**

The "Review a Paper" feature in a Conference Management System allows assigned reviewers to evaluate and provide detailed feedback on submitted papers. This functionality simplifies the peer-review process, ensuring that submissions are assessed based on established criteria, such as originality, relevance, methodology, and clarity. It promotes efficient communication between reviewers and organizers while maintaining confidentiality and transparency, especially in blind review settings.

**2. Actor**

**2.1. Primary actor** :Reviewer

**2.2. Secondary Actor**: None

**3. Preconditions**

* The reviewer must be logged in and be authorized to perform this use case.
* A review scenario is opened through a Make review operation.
* The state of the Conference must be “Review period started” in order to perform a review. The review cannot be performed if the conference is in “Submission are available” state.

4. **Postconditions**

* Conference database should remain in a consistent state
* Research Paper database should remain in a consistent state.

**5. Basic Flow**

1. This use case starts when the actor wants to create a review
2. Reviewer proceeds to the assigned papers page
3. The system checks the conference timeline.
4. The reviewer is able to view assigned papers.
5. The user exits the system

**6. Alternative Flows**

This alternative flow applies when the review period has not started yet

This flow can occur after step 2

2.1. The use case ends.

**7. Special Requirements**

None

**Use Case 5**

**Use case name** : Check Status/feedback

**1. Introduction**

The "Check Status/Feedback" functionality in the Conference Management System enables authors to monitor the status of their submitted papers and access feedback provided by reviewers or conference organizers. This feature promotes transparency in the review process and supports authors in making necessary revisions.

**2. Actor**

**2.1. Primary actor** :Author

**2.2. Secondary Actor**: Reviewer

**3. Preconditions**

* The author must be logged in and be authorized to perform this use case
* The author must have submitted atleast one research paper.

**4.** **Postconditions**

* Project database should not suffer any change if only save (not commit) operations were executed with success on scenarios.

**5. Basic Flow**

This use case starts when the actor wants to check status of an entry for the conference

1. Author proceeds to the profile page
2. The author proceeds to the tab that contains the submissions information
3. The system will display the submissions information and feedback from reviewer.
4. The user checks the status of its submission
5. The user exits the system

**6. Alternative Flows**

This alternative flow applies if the system halts

This flow can occur at any step

1. The author logs in again in the system, and goes to the profile page

**7. Special Requirements**

None

**Use Case 6**

**Use case name** : Logout

**1. Introduction**

The "Logout" functionality in a Conference Management System enables users to safely exit the platform once their tasks are completed. It ensures that their session is fully terminated, safeguarding sensitive data from unauthorized access. This feature is essential for ensuring privacy and security, particularly when users access the system on shared or public devices. After logging out, users are redirected to the login page, requiring them to re-authenticate in order to regain access to the system.

**2. Actor**

**2.1. Primary actor** : Chair,Author, Reviewer

**2.2. Secondary Actor**: None

**3. Preconditions**

* User must be logged in

**4. Postconditions**

* The user’s session is terminated.
* The user is redirected to the login or homepage.

**5. Basic Flow**

1. The user clicks the "Logout" button/link.
2. The system clears all locally stored session data, including cookies, tokens, and cache.
3. The user is redirected to the login page or homepage with a confirmation message (e.g., "You have been logged out Successfully").

**6. Alternative Flows**

None

**7. Special Requirements**

None

**Use Case 7**

**Use case name** : Submit Feedback

**1. Introduction**

The "Submit Feedback" feature in a Conference Management System (CMS) allows reviewers to provide detailed evaluations of submitted papers. Reviewers assess papers based on predefined criteria such as originality, relevance, clarity, and technical quality. This feature ensures that feedback is structured, comprehensive, and delivered efficiently to authors, helping them refine their work. The process also aids conference organizers in determining which submissions meet the standards for acceptance.

**2. Actor**

**2.1. Primary actor** : Reviewer

**2.2. Secondary Actor**: Author

**3. Preconditions**

* Reviewer must be logged in.
* Reviewer must be authorised to perform this operation
* Conference must be in “Review period has started” state .

**4. Postconditions**

* Review information is stored in database against selected paper
* Author associated with the paper is notified of the updated status and feedback.
* Research paper database must remain in a consistent state.

**5. Basic Flow**

1. The Reviewer navigates to the “Submit feedback" section.
2. The system checks the conference timeline.
3. The system displays a list of submitted papers with their current status.
4. The Reviewer selects a paper to submit feedback.
5. The system presents a form with the current status and a dropdown or input field for a new status.
6. The Reviewer enters the new status and confirms the update.
7. The system validates the new status and updates the database.
8. The system notifies the associated Author of the updated status via email or in-app notification.

**6. Alternative Flows**

This alternative flow applies when the reviews are not opened yet

This flow can occur after step 2

2.1. The use case ends

**7. Special Requirements**

None

**Use Case 8**

**Use case name** : Submit revised Paper

**1. Introduction**

The "Submit Revised Paper" feature in  Conference Management System allows authors to upload updated versions of their papers in response to reviewer feedback.

**2. Actor**

**2.1. Primary actor** : Author

**2.2. Secondary Actor**: None

**3. Preconditions**

* The author must be logged in and be authorized to perform this use case
* A submission scenario is opened through an Submit revised paper operation
* The state of the Conference must be “Review period started” in order to perform a submission.
* Atleast one paper must have been submitted by the Author.
* Paper submitted by the Author must have been suggested changes by the Reviewer.

**4. Postconditions**

* The submission information is saved in the application database
* Conference database should remain in a consistent state if the Save/Commit operation has ended with success.
* Conference entities should not suffer any changes if the Save/Commit operation has not ended with success.
* Reasearch Paper database should not suffer any change if only save (not commit) operations were executed with success on scenario.

**5. Basic Flow**

This use case starts when the actor wants to create/modify an entry for the conference

1. Author proceeds to the Revised paper submission.
2. The system checks the conference timeline .
3. The author proceeds to the page that contains the submission form .
4. The author enters the credentials for the submission (e.g. Name, Keywords) .
5. The author selects the file he/she wants to upload as paper .
6. The system will indicate that the revised submission has been registered to the system.
7. The user exits the system .

**6. Alternative Flows**

This alternative flow applies when the Author is not eligible to submit revised paper

This flow can occur after step 2

2.1. The use case ends

This alternative flow applies if the paper is not uploaded

This flow can occur after step 5

5.1 The author re-enters the information

5.2 The system will validate the information

5.4 The use case continues with step

**7. Special Requirements**

None

**Use Case 9**

**Use case name** : Assign Reviewers

**1. Introduction**

The "Assign Reviewers" feature in a Conference Management System enables program chairs to allocate submitted papers to suitable reviewers based on their expertise. The system simplifies reviewer management, balances workloads, and ensures papers are reviewed by qualified experts.

**2. Actor**

**2.1. Primary actor** : Chair

**2.2. Secondary Actor**: Reviewer

**3. Preconditions**

* The conference chair must be logged in
* An assignment scenario is opened through an Assign Papers operation
* The state of the Conference must be “Review period has started” in order to assign. The assignment cannot be performed if the conference is in “Submission are opened” or “Review period not started”

**4. Postconditions**

* Project database should not suffer any change if only save (not commit) operations were executed with success on scenarios.

**5. Basic Flow**

This use case starts when the Chair wants to assign reviewers for each paper

1. Conference chair proceeds to homepage
2. The system checks the conference timeline
3. The program chair starts an assignment scenario
4. The system checks bag of words computation for the reviewer papers
5. The system checks bag of words computation for the author papers
6. The system checks for similarities
7. The system checks for conflict of interest
8. The paper is assigned to reviewer
9. The user will exit the system

**6. Alternative Flows**

This alternative flow applies when the review period has not yet started

This flow can occur after step 2 .

2.1. The use case ends

This alternative flow applies if bag of words for the reviewer has been not computed

This flow can occur after step 4

4.1 The system computes the bag of words for reviewer

4.2 The use case continues with step 5

This alternative flow applies if bag of words for the author has been not computed

This flow can occur after step 5

5.1 The system computes the bag of words for author

5.2 The use case continues with step 6

This alternative flow applies if there is a conflict of interest based on bag of words

This flow can occur after step 7

7.1 The system computes the bag of words

7.2 The use case continues with step 6

This alternative flow applies if there are papers that don’t have reviewers

This flow can occur after step 9

9.1 The system checks if there are papers that are not assigned

9.2 The use case continues with step

**7. Special Requirements**

None

**Use Case 10**

**Use case name** : Manage Conference

**1. Introduction**

The "Manage Conference" feature in a Conference Management System (CMS) enables organizers to oversee and manage all aspects of a conference. This includes creating and editing conference details, managing schedules, monitoring participant registrations, handling submissions, assigning reviewers, and communicating with attendees. The feature ensures that organizers have full control over the conference workflow to ensure its successful execution.

**2. Actor**

**2.1. Primary actor** : Chair

**2.2. Secondary Actor**: None

**3. Preconditions**

* The program Chair must be logged in and be authorized to perform this use case
* A conference management scenario is opened through an Manage Conference operation

**4. Postconditions**

* Project database should not suffer any change if only save (not commit) operations were executed with success on scenarios.

**5. Basic Flow**

1. The program chair chooses one of the following operations
   1. “Add conference”
   2. “View conference”
2. If the program chair chooses “Add conference” then add conference sub-flow is executed.
3. If program chair chooses “view conference” then view conference sub-flow is executed.
   1. **Add conference**
4. Program chair proceeds to the conference submission form
5. The program chair enters the identification data for the conference (e.g. Name, Deadlines)
6. The system checks the data
7. The system will indicate that the conference information has been registered to the system
8. The user exits the system
   1. **View conference**
9. The program chair selects a conference from a list of all ongoing conferences
10. A list of papers submitted in that conference is shown along with its current status(i.e, accepted, rejected, changes proposed).

**6. Alternative Flows**

This alternative flow applies if the system halts

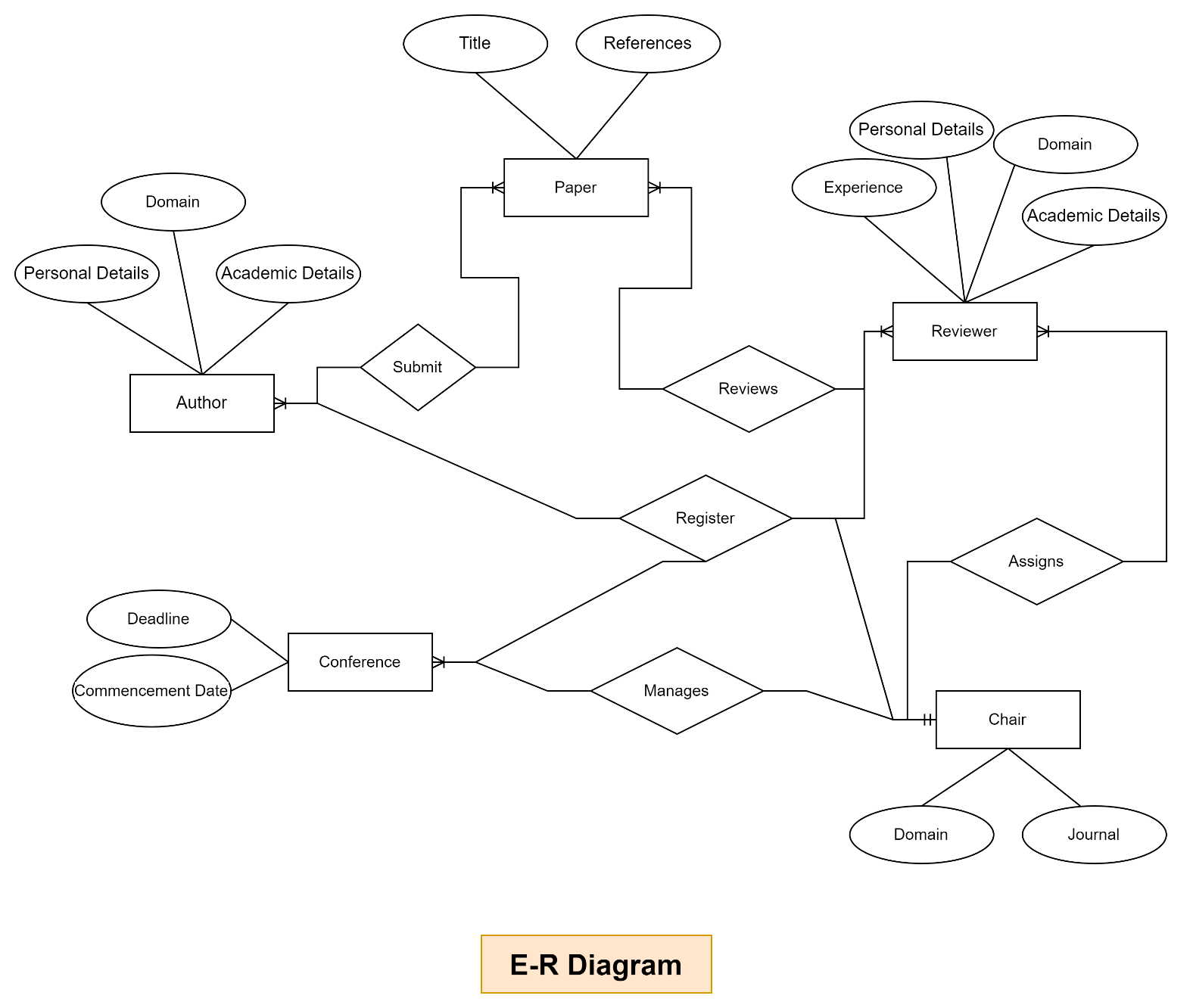
This flow can occur at any step

1. The program chair logs in again in the system, and goes to the add conference page

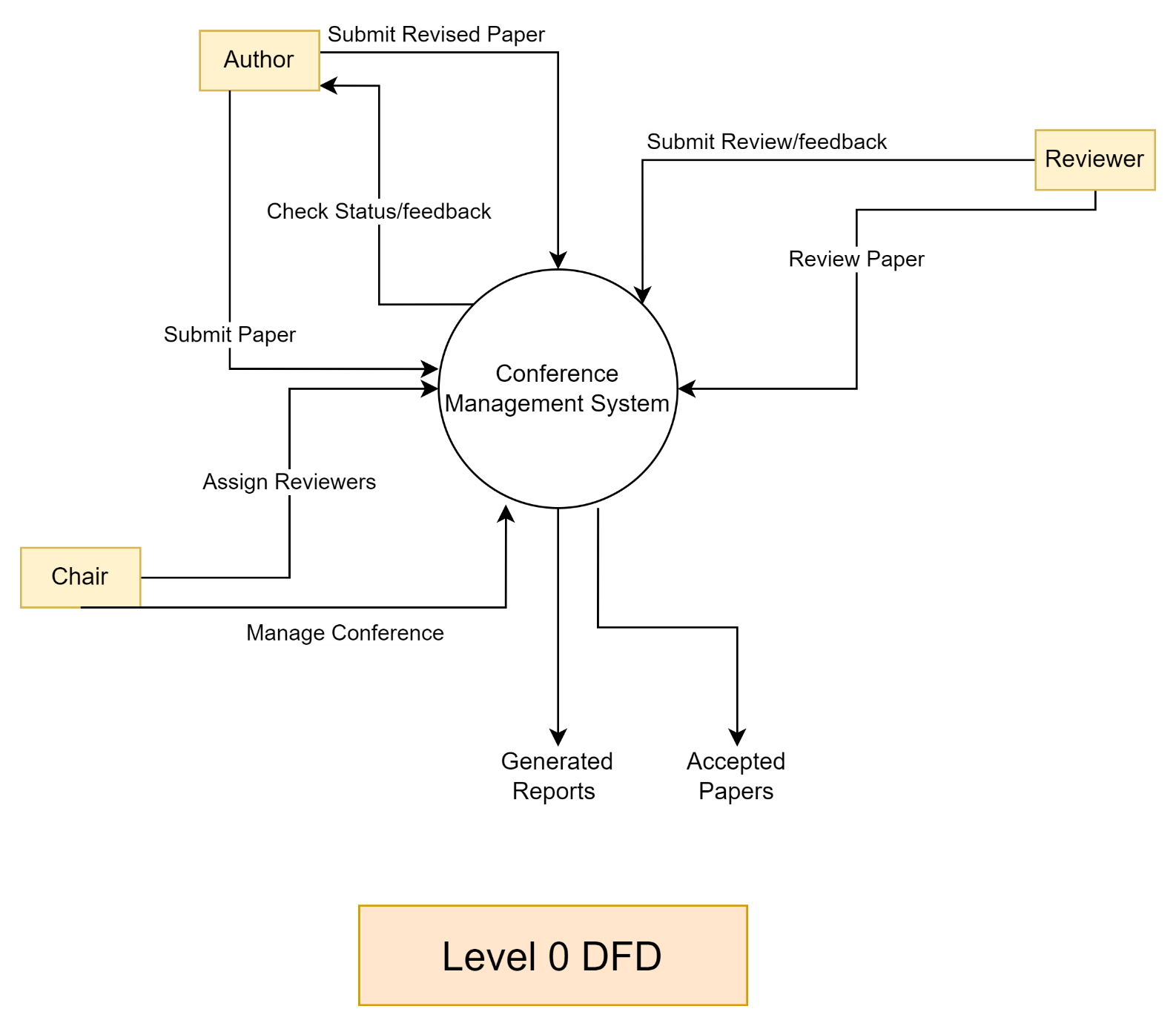
**7. Special Requirements**

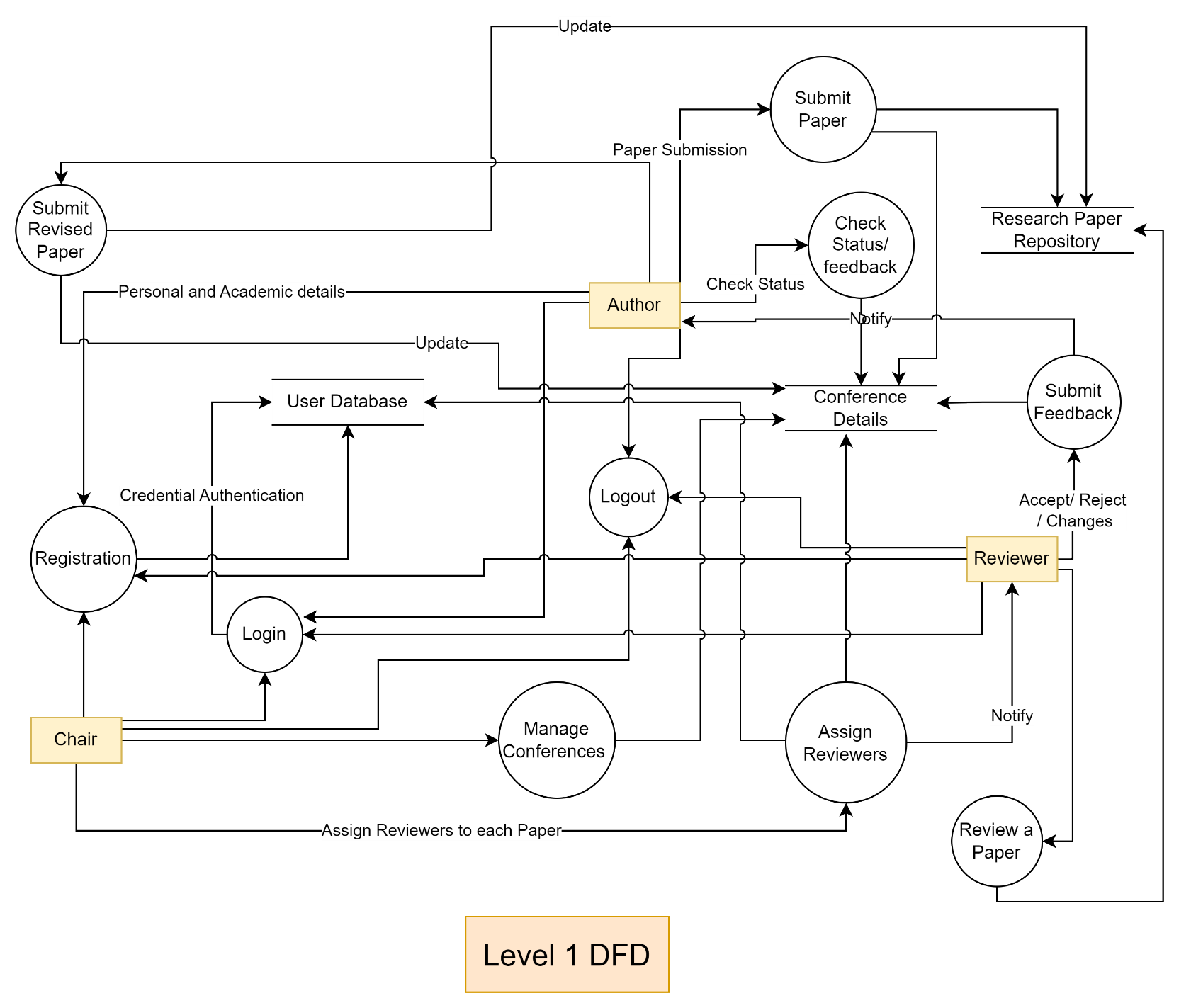
None

**ER DIAGRAM**

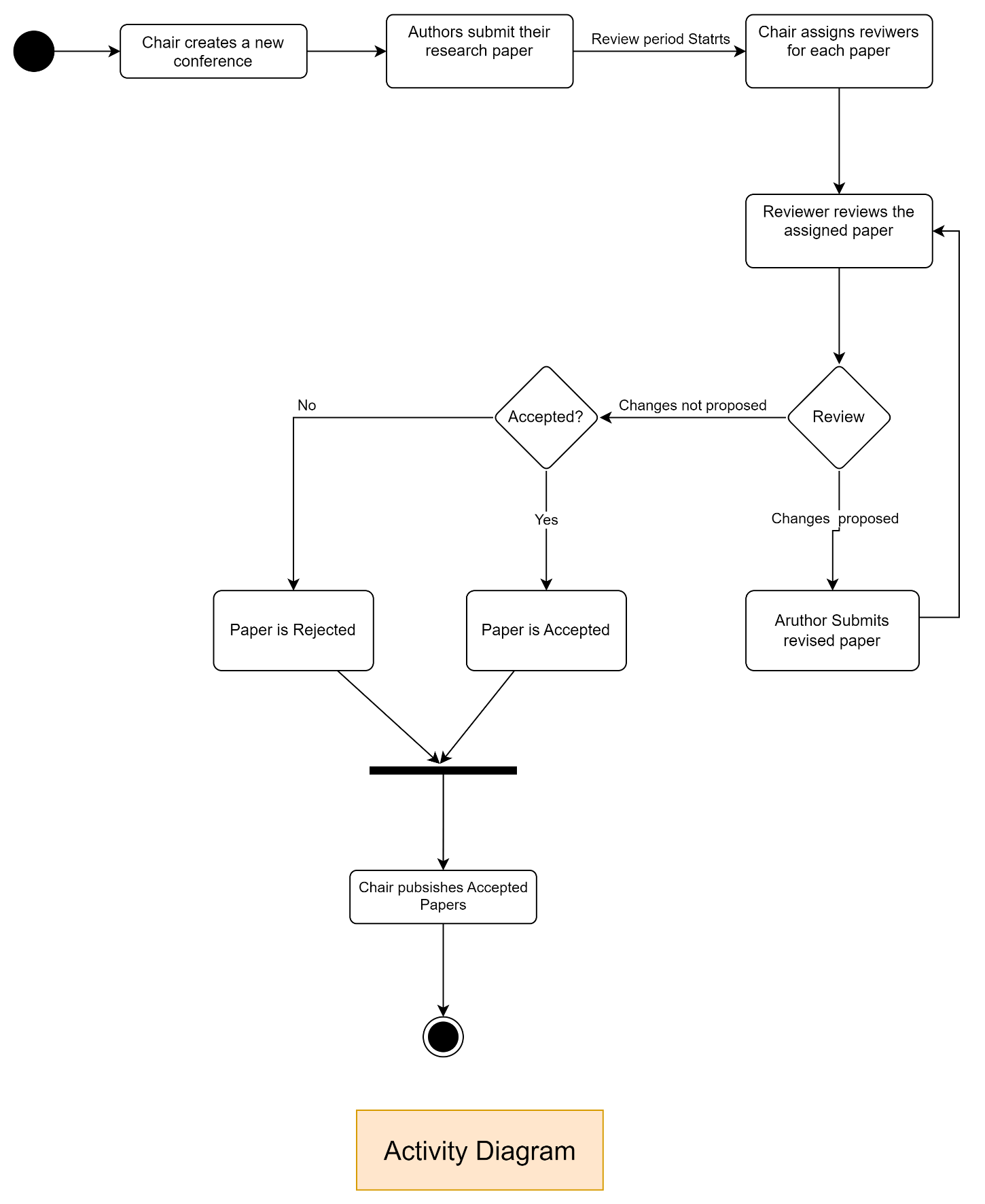


**DATA FLOW DIAGRAMS**

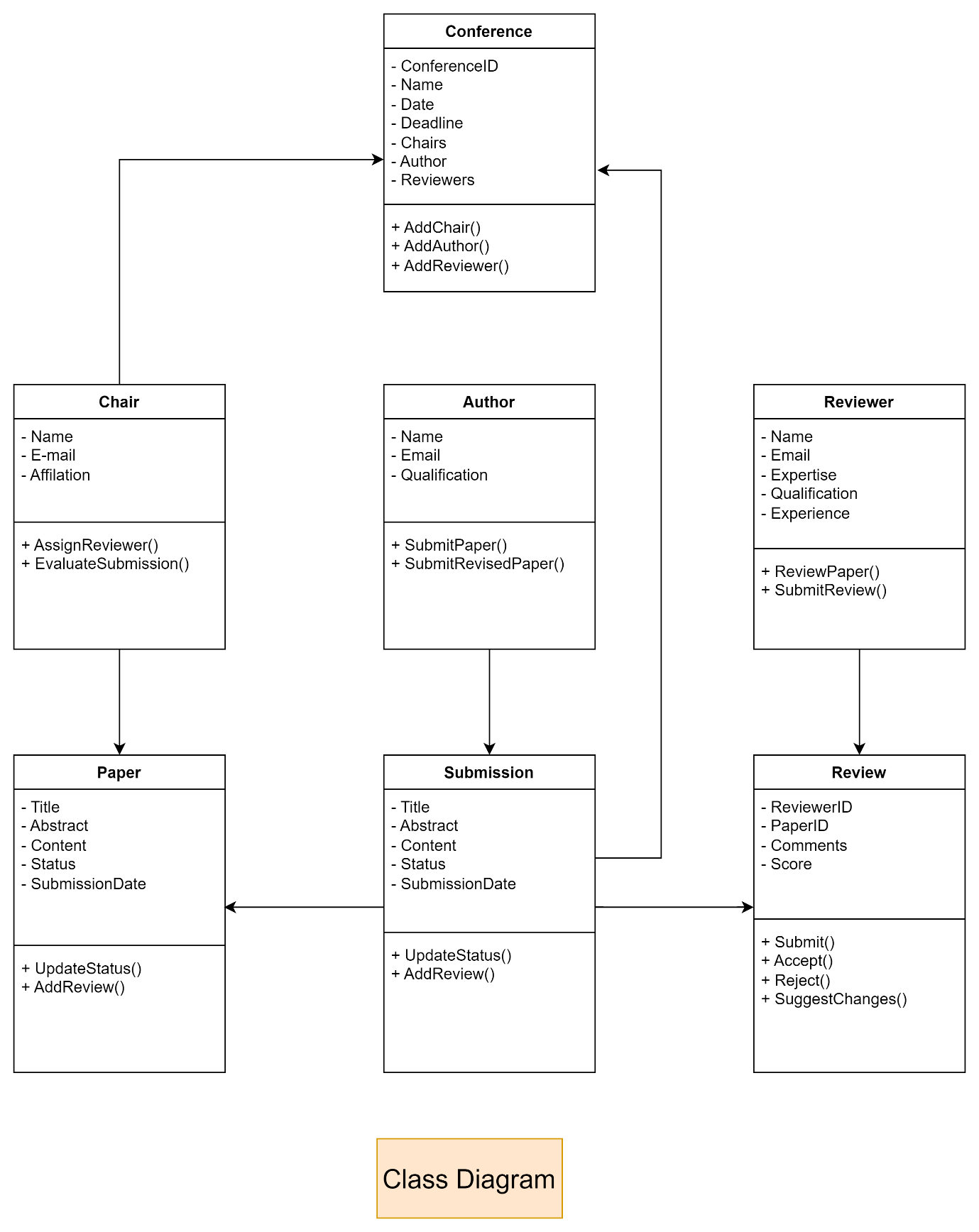




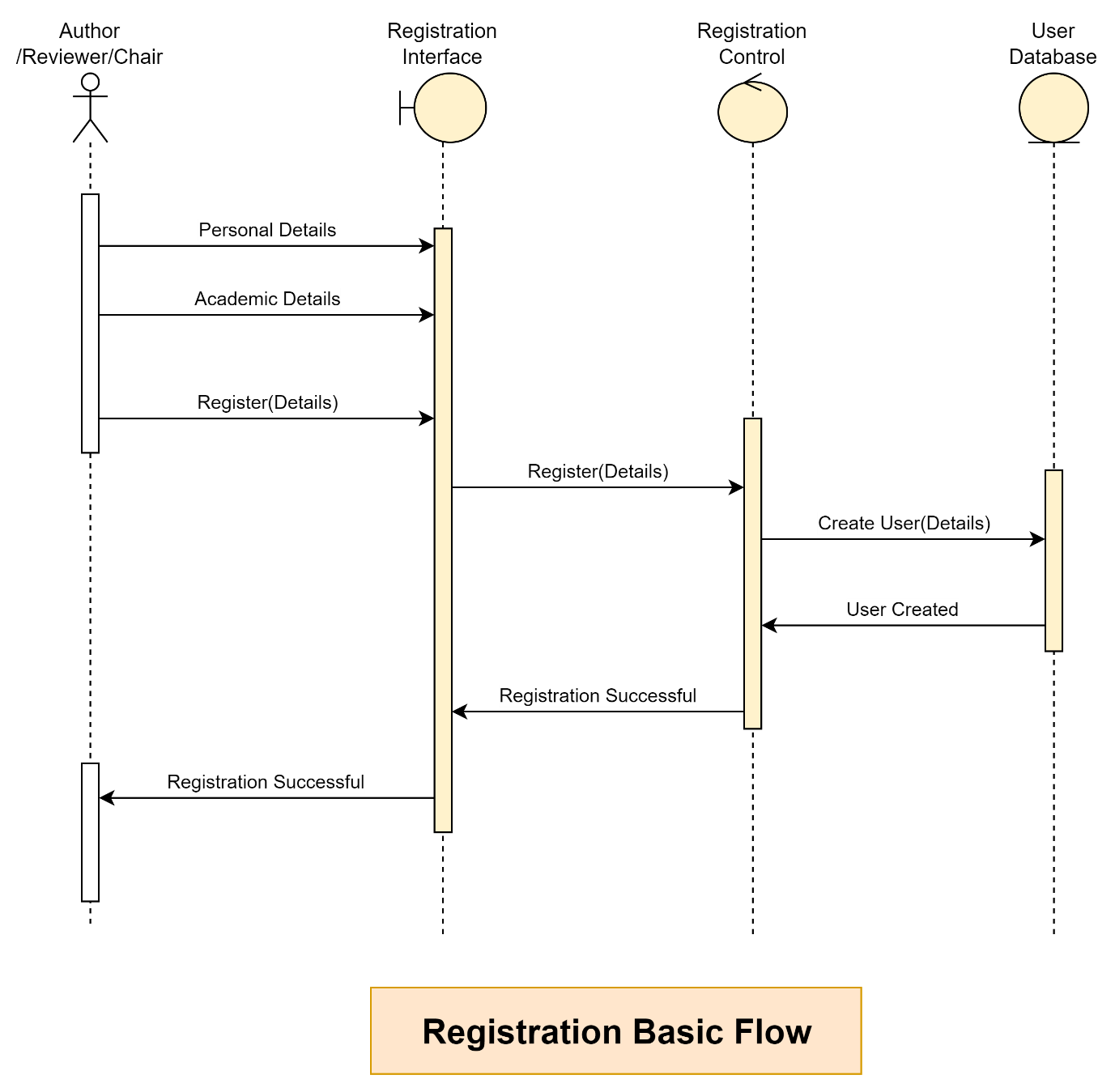
**ACTIVITY DIAGRAM**

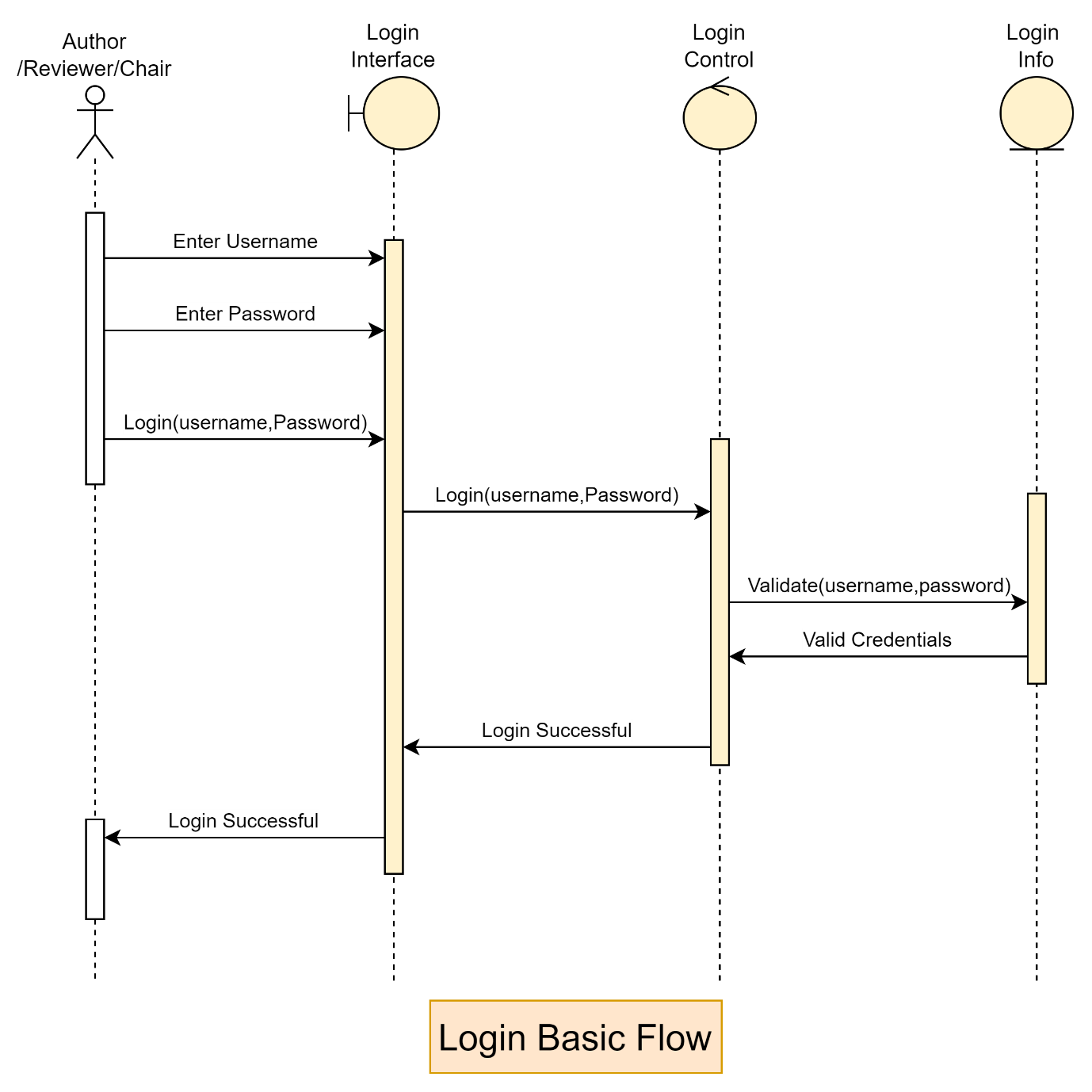


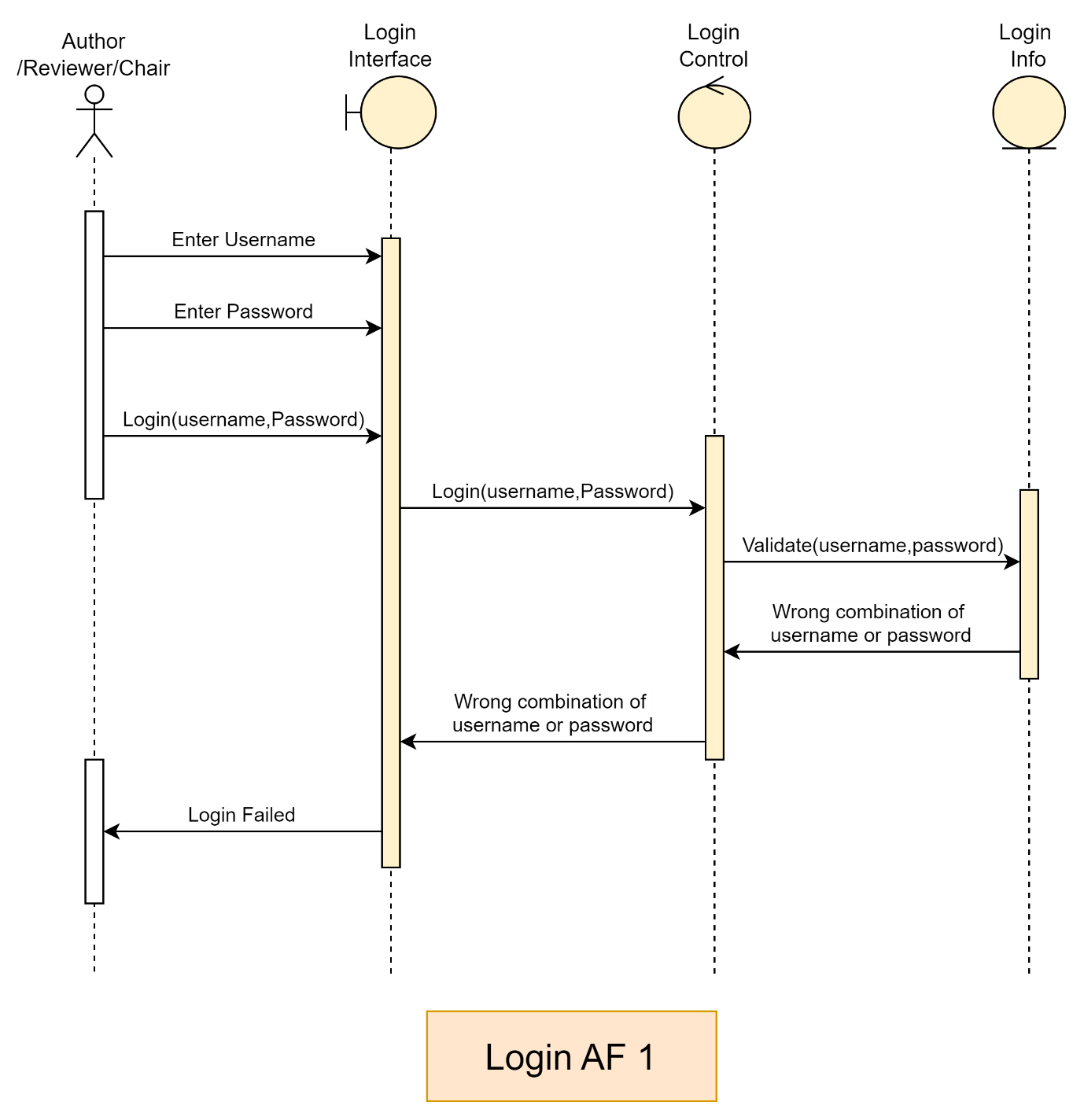
**CLASS DIAGRAMS**

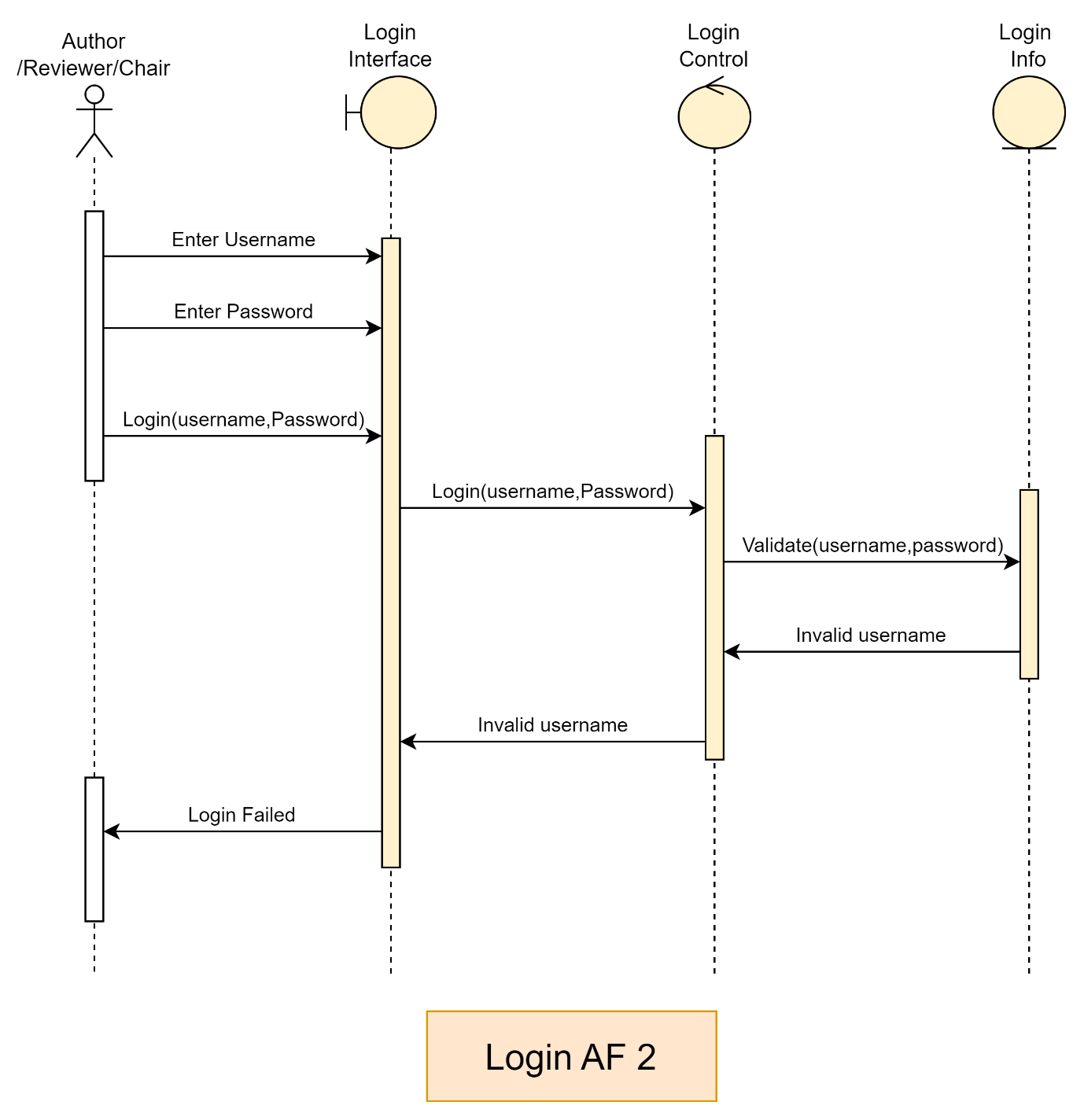


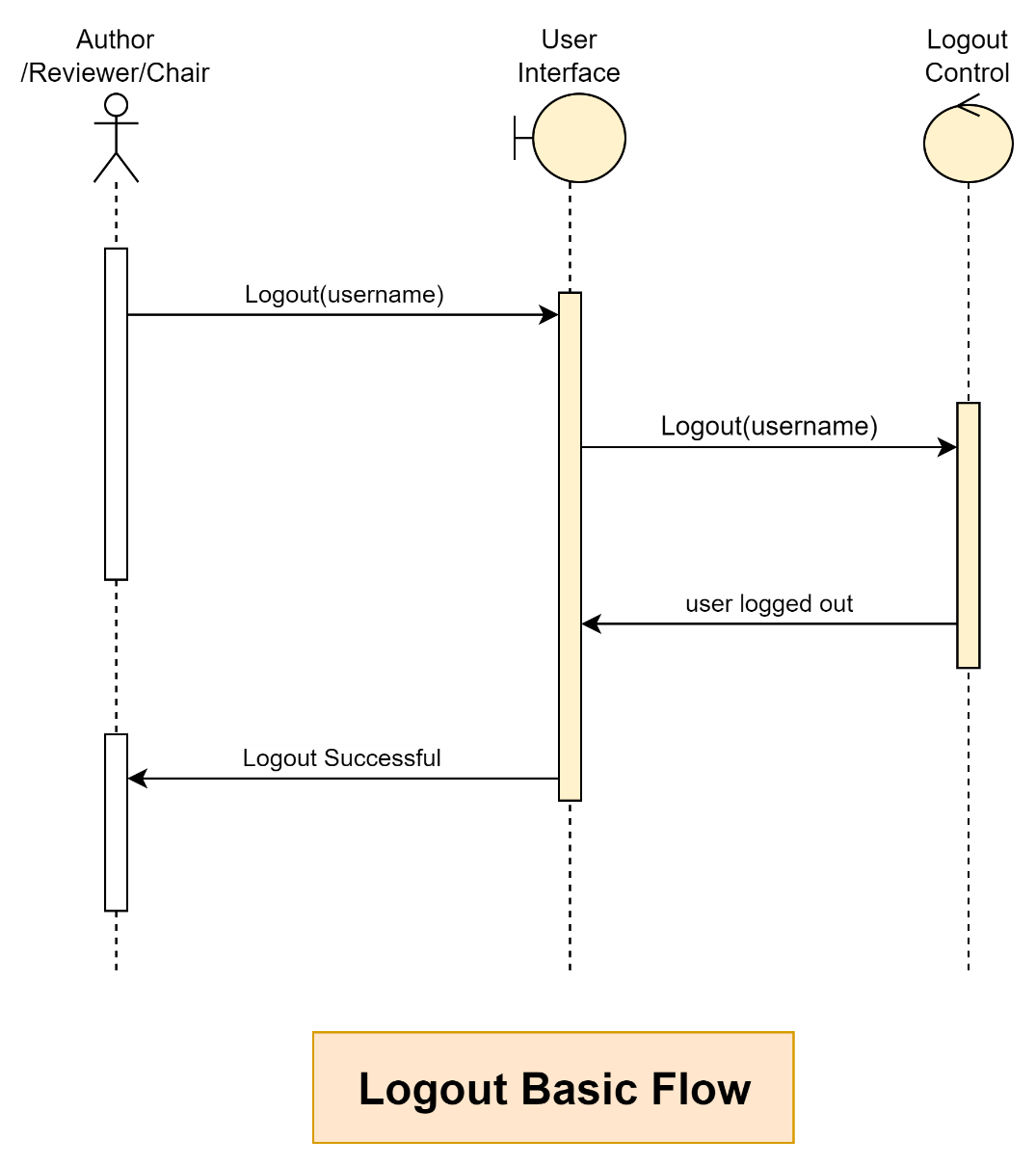
**SEQUENCE DIAGRAMS**

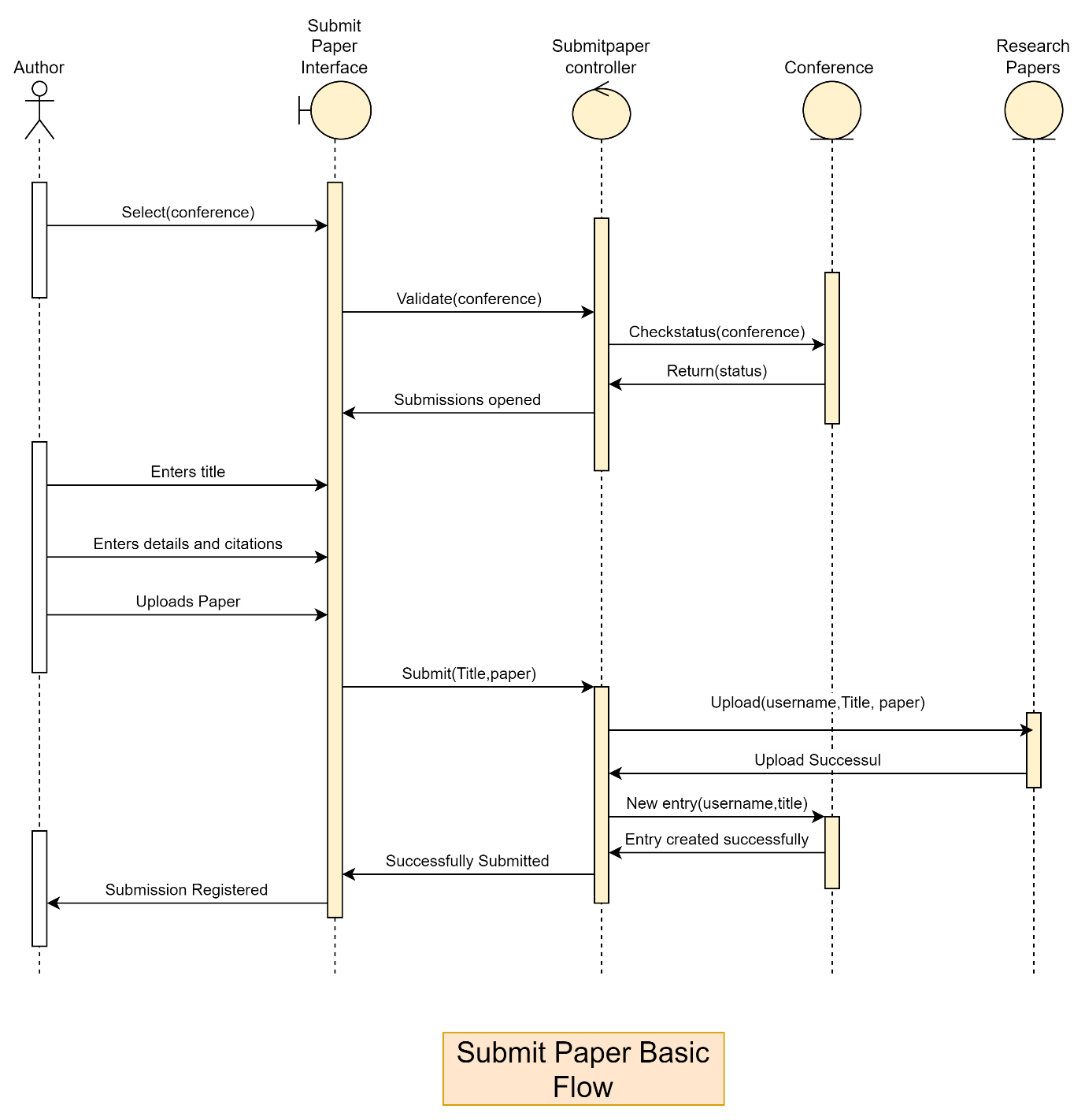


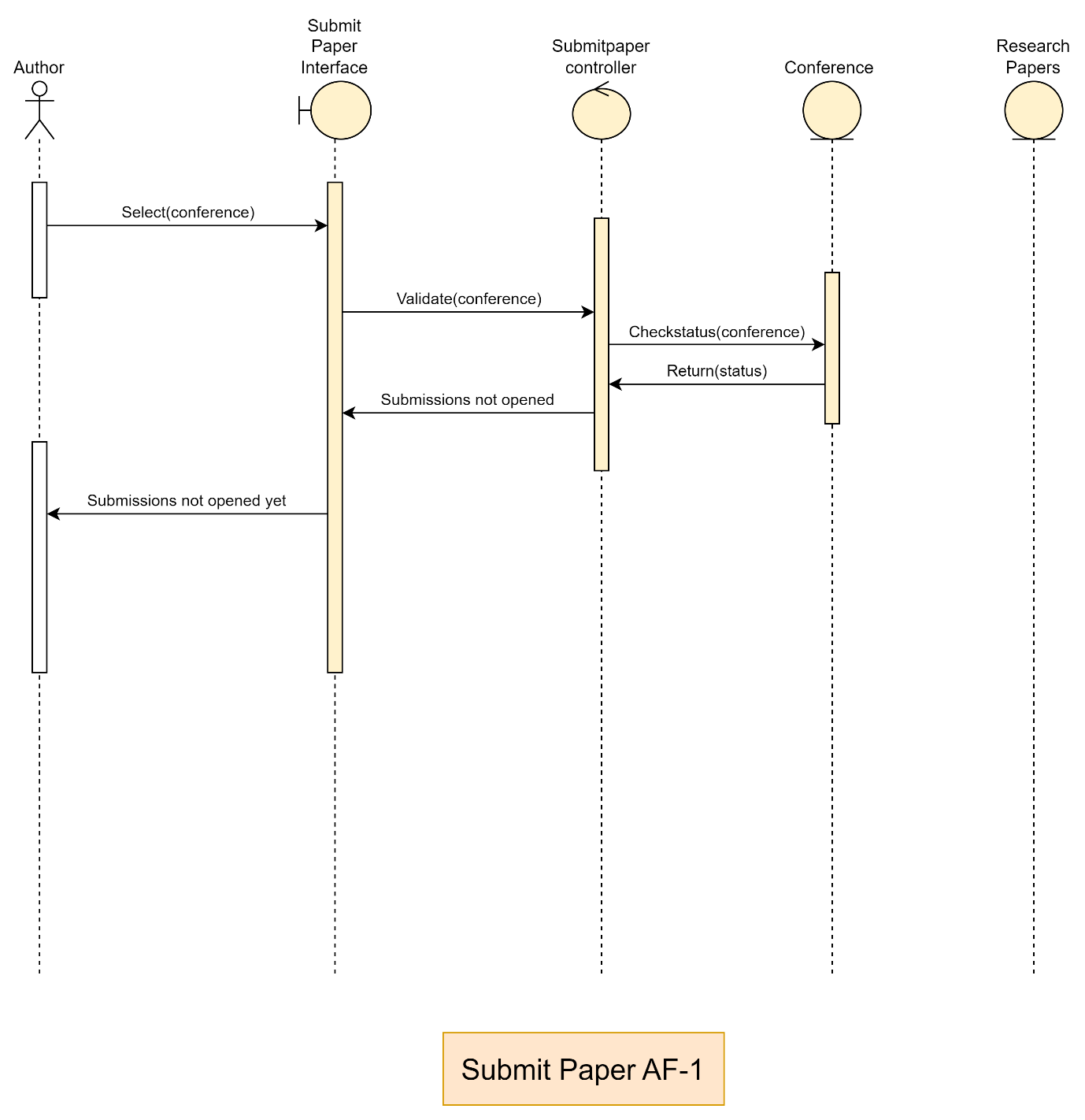


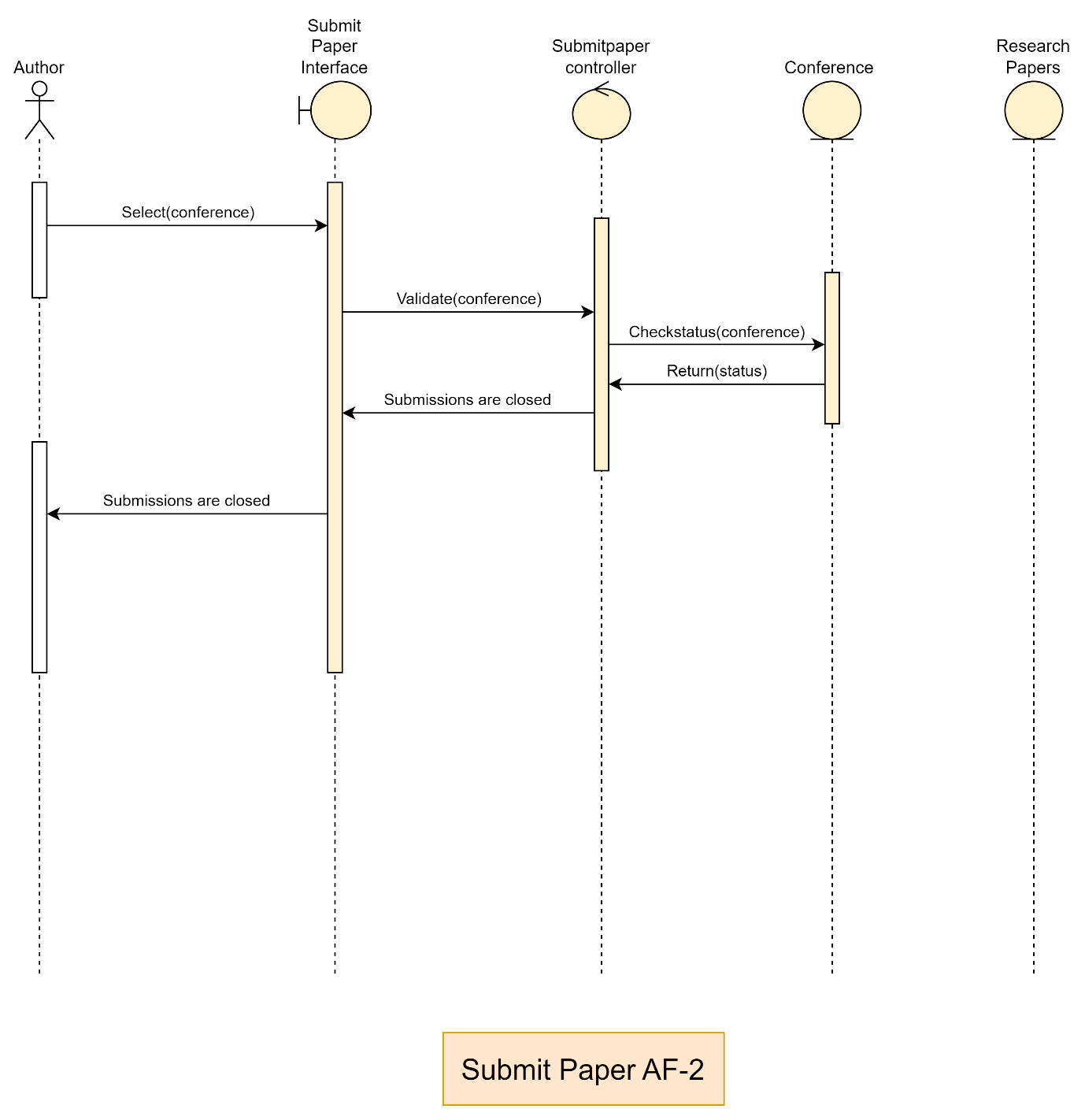


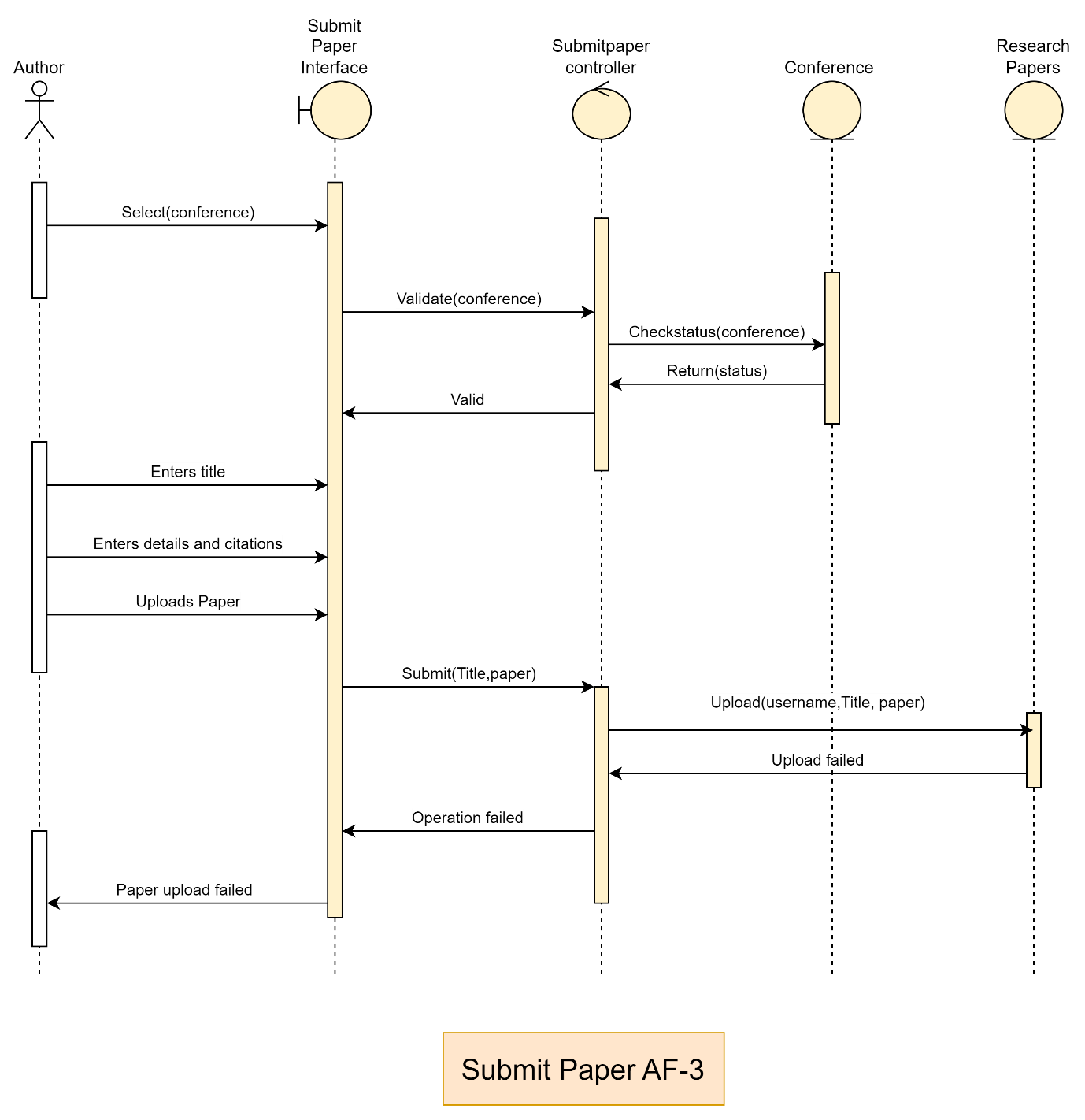


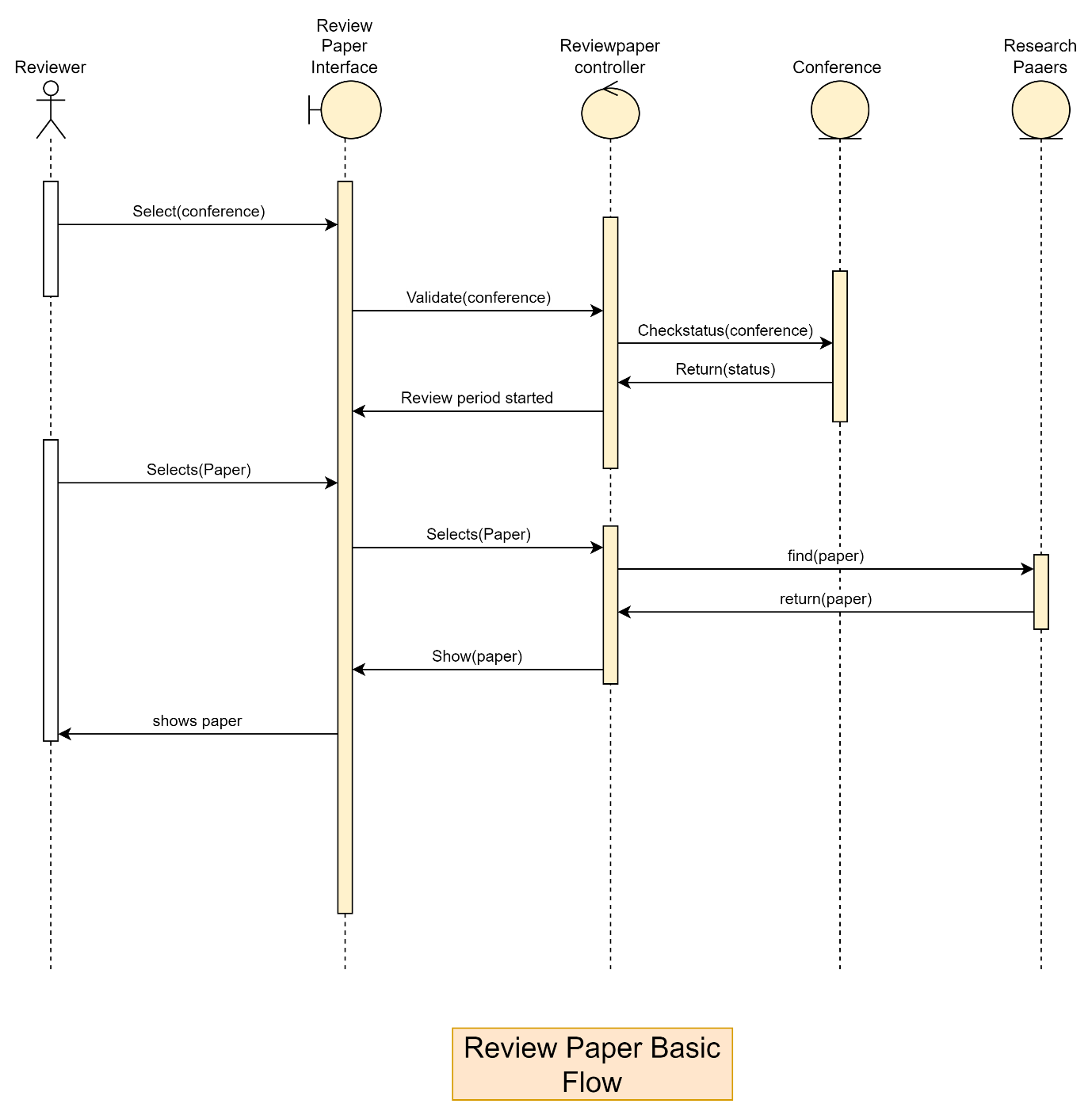


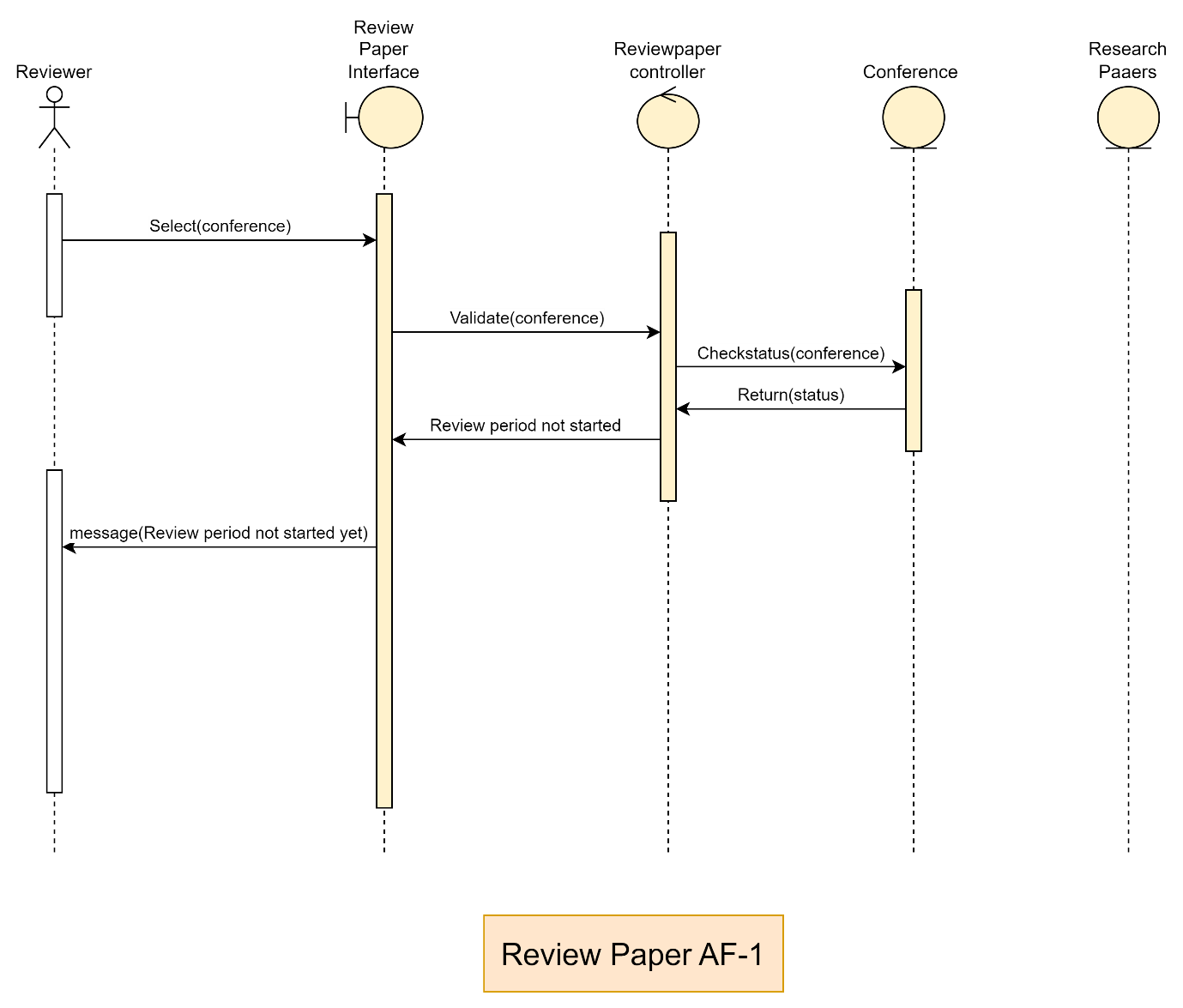


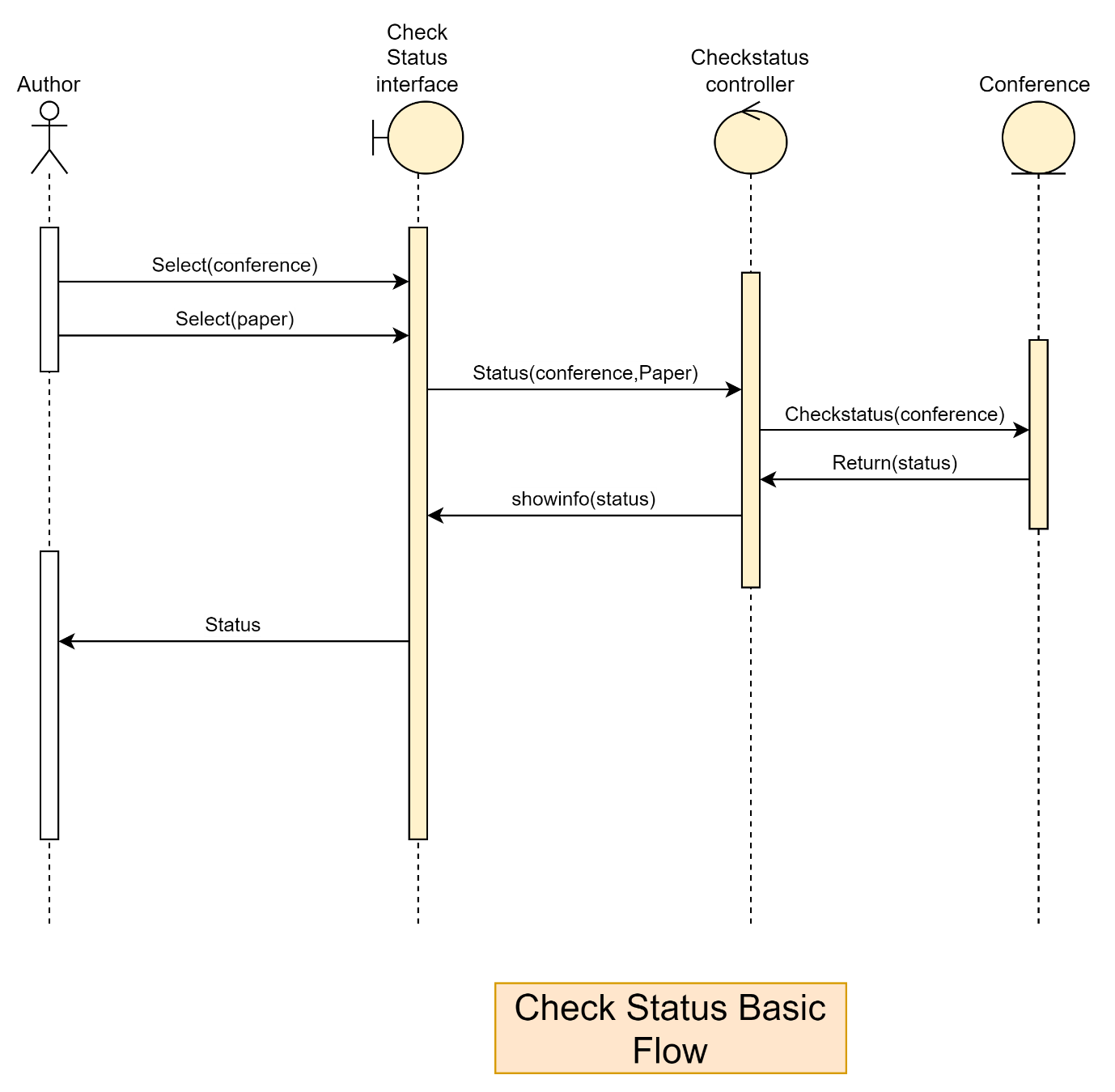


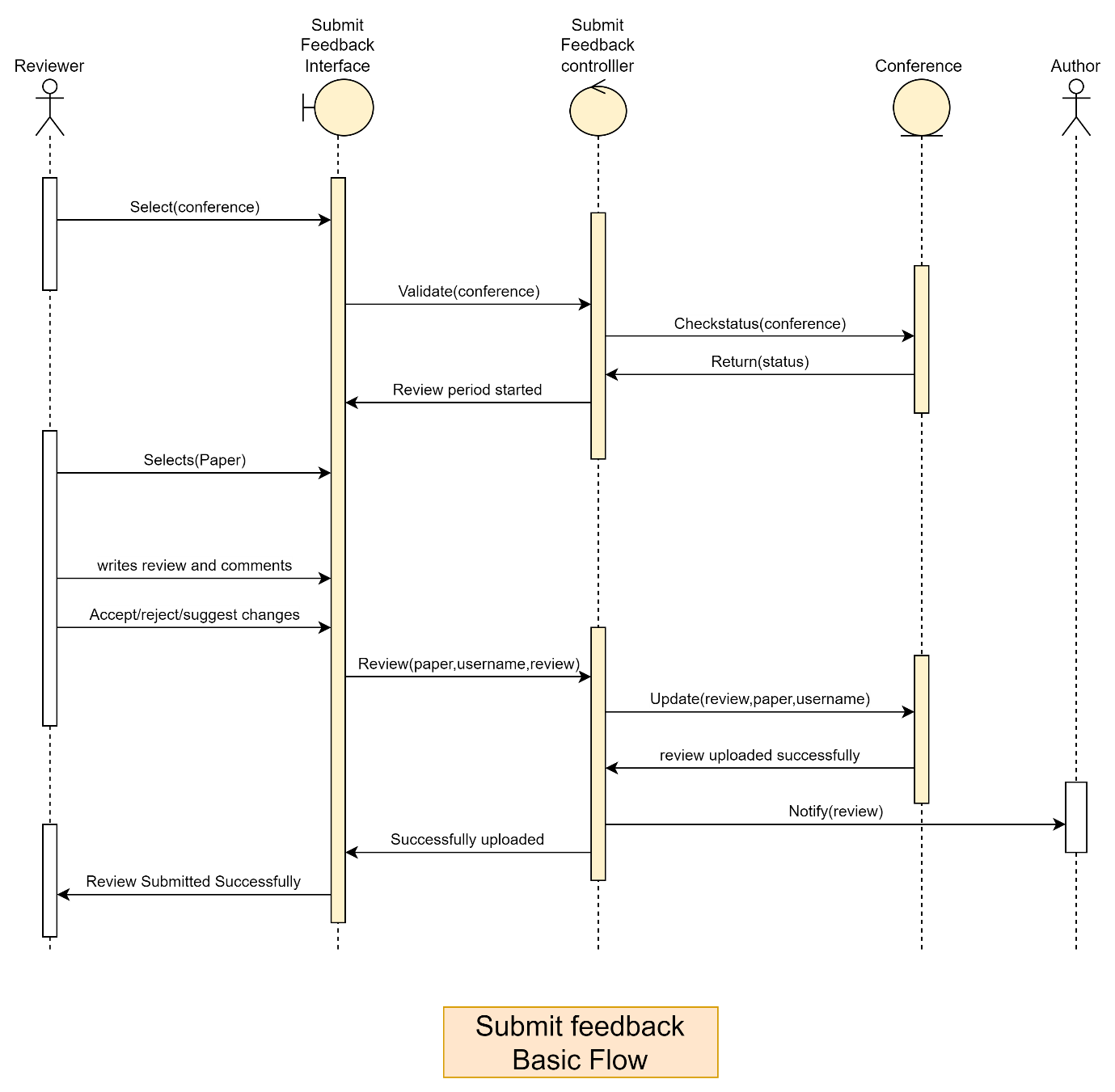


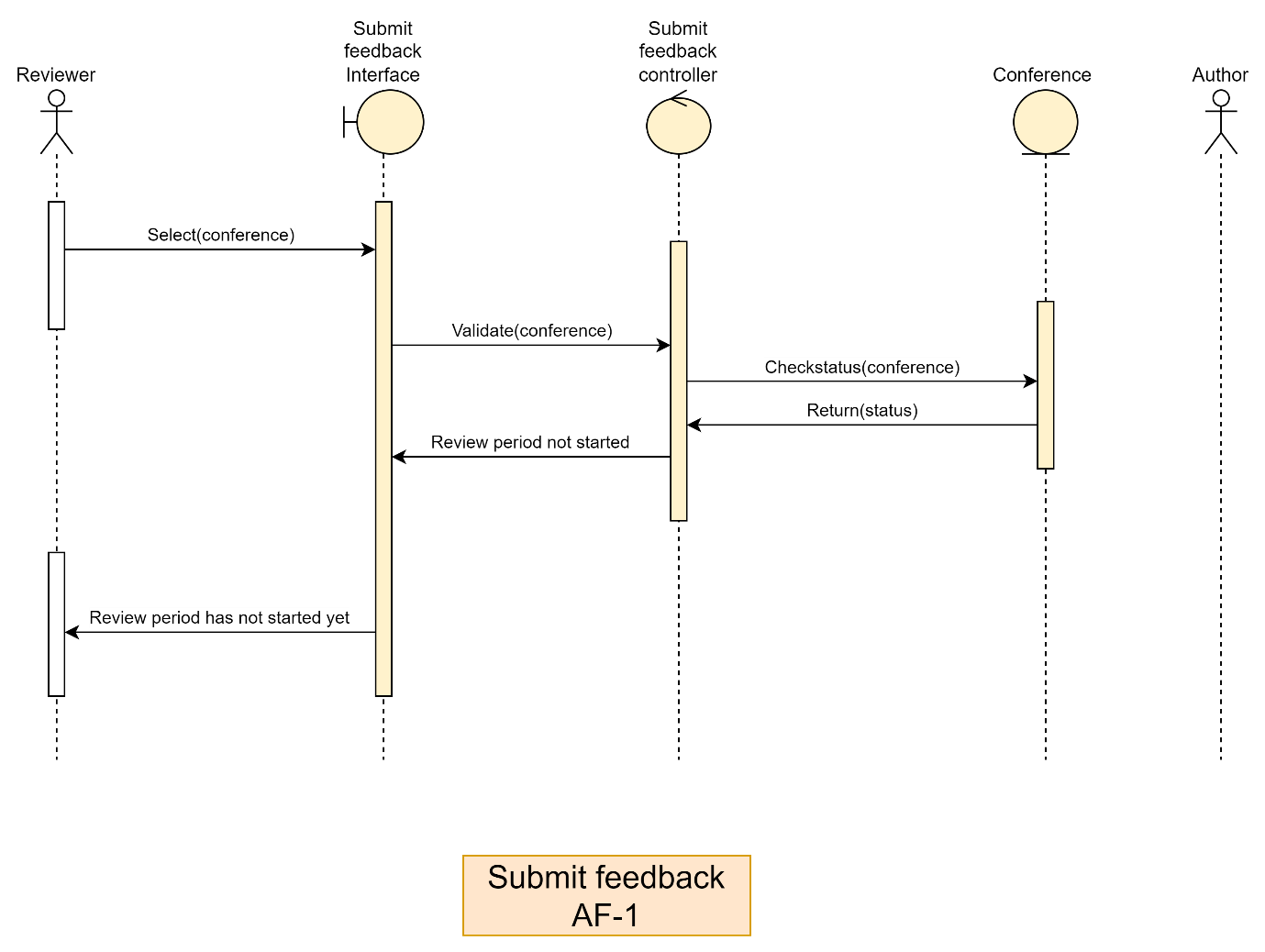


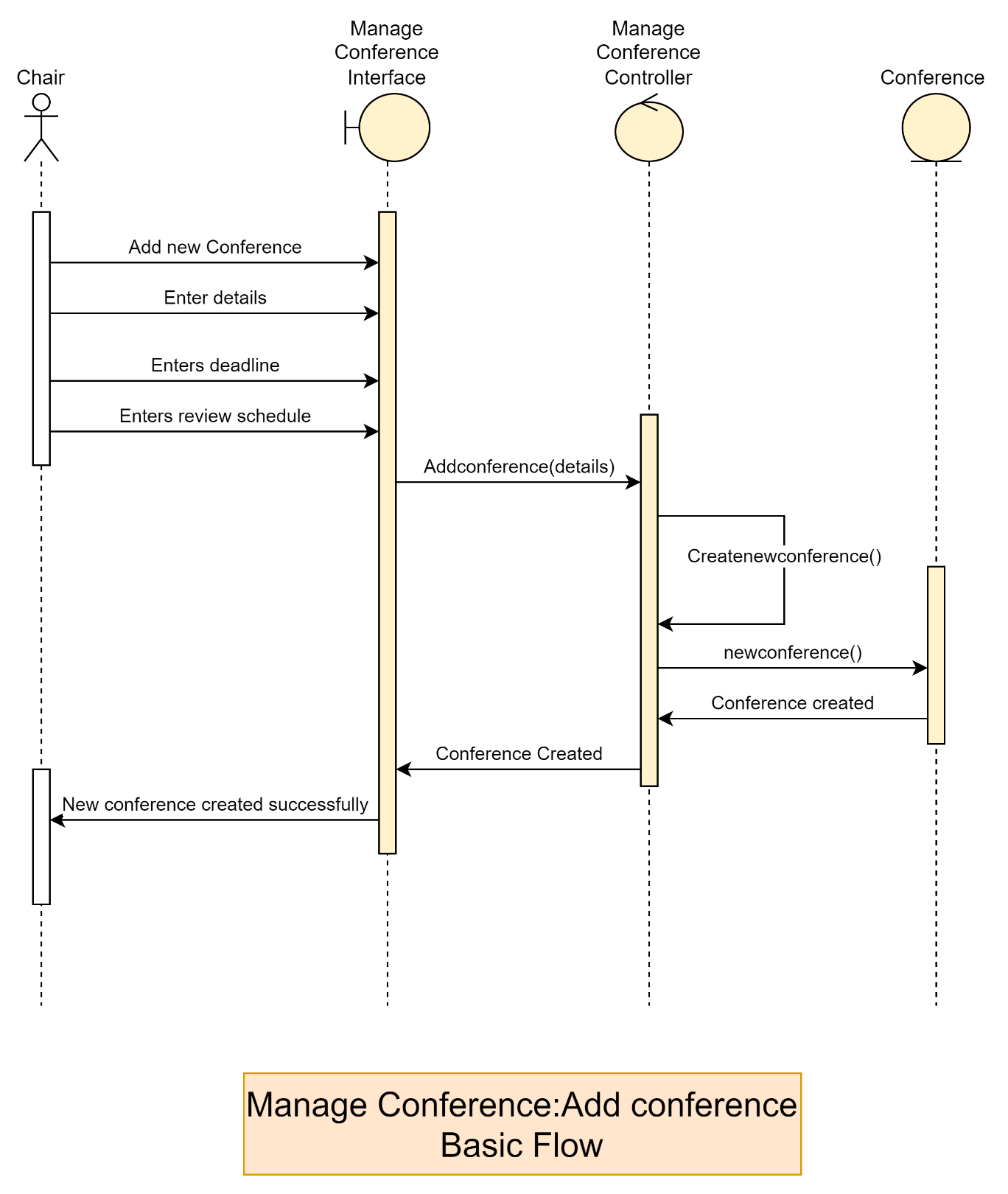


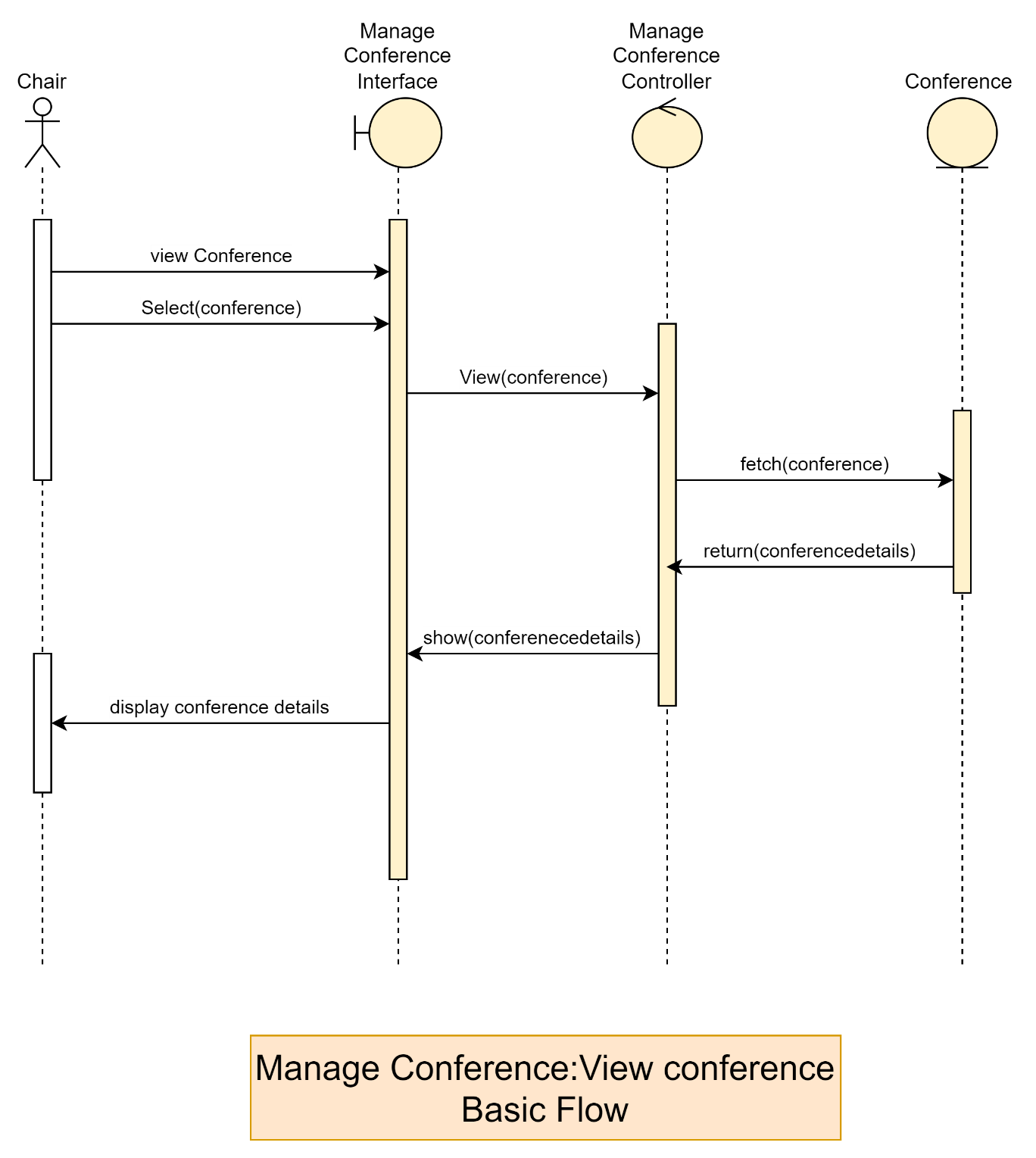


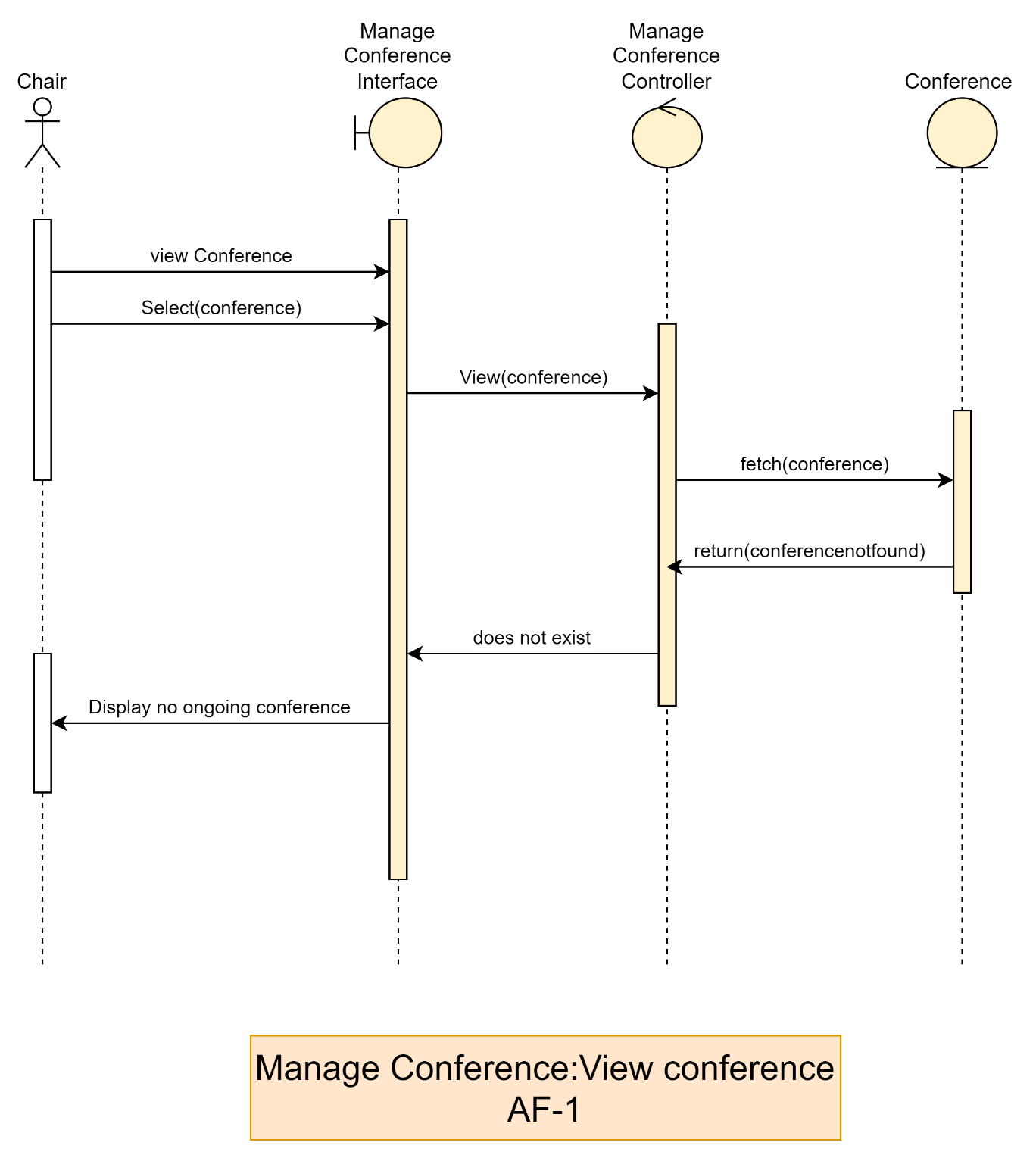


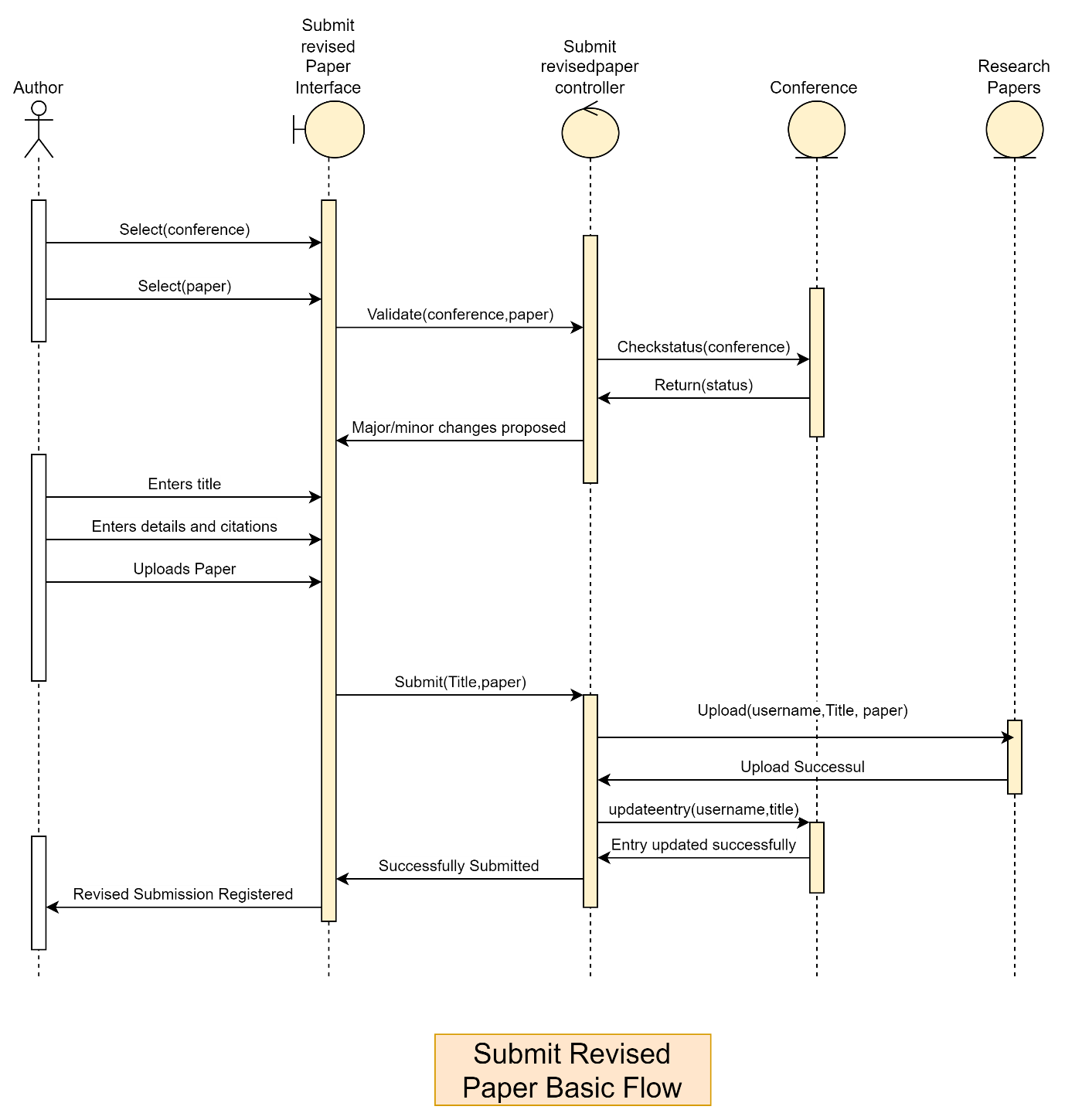


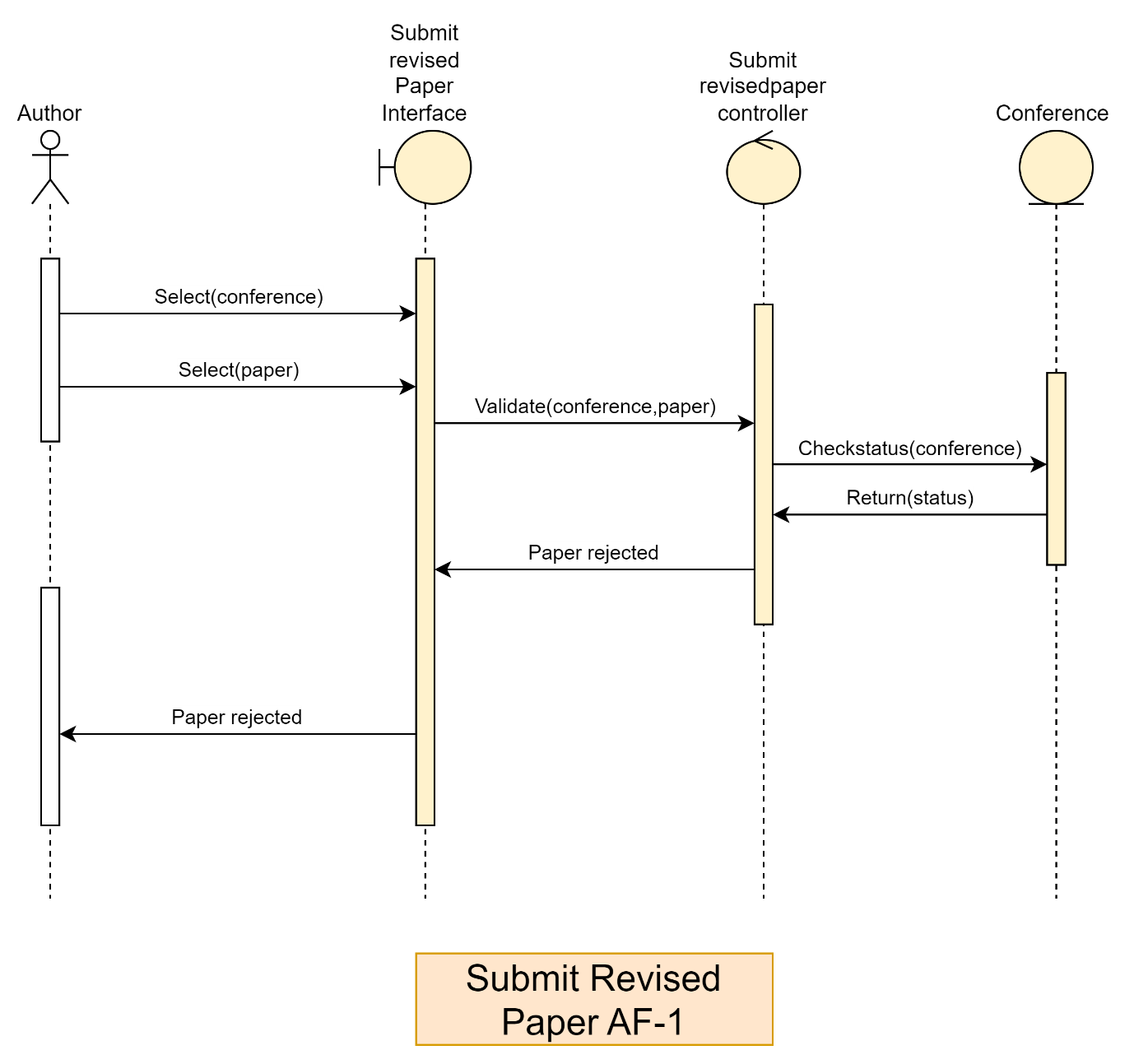


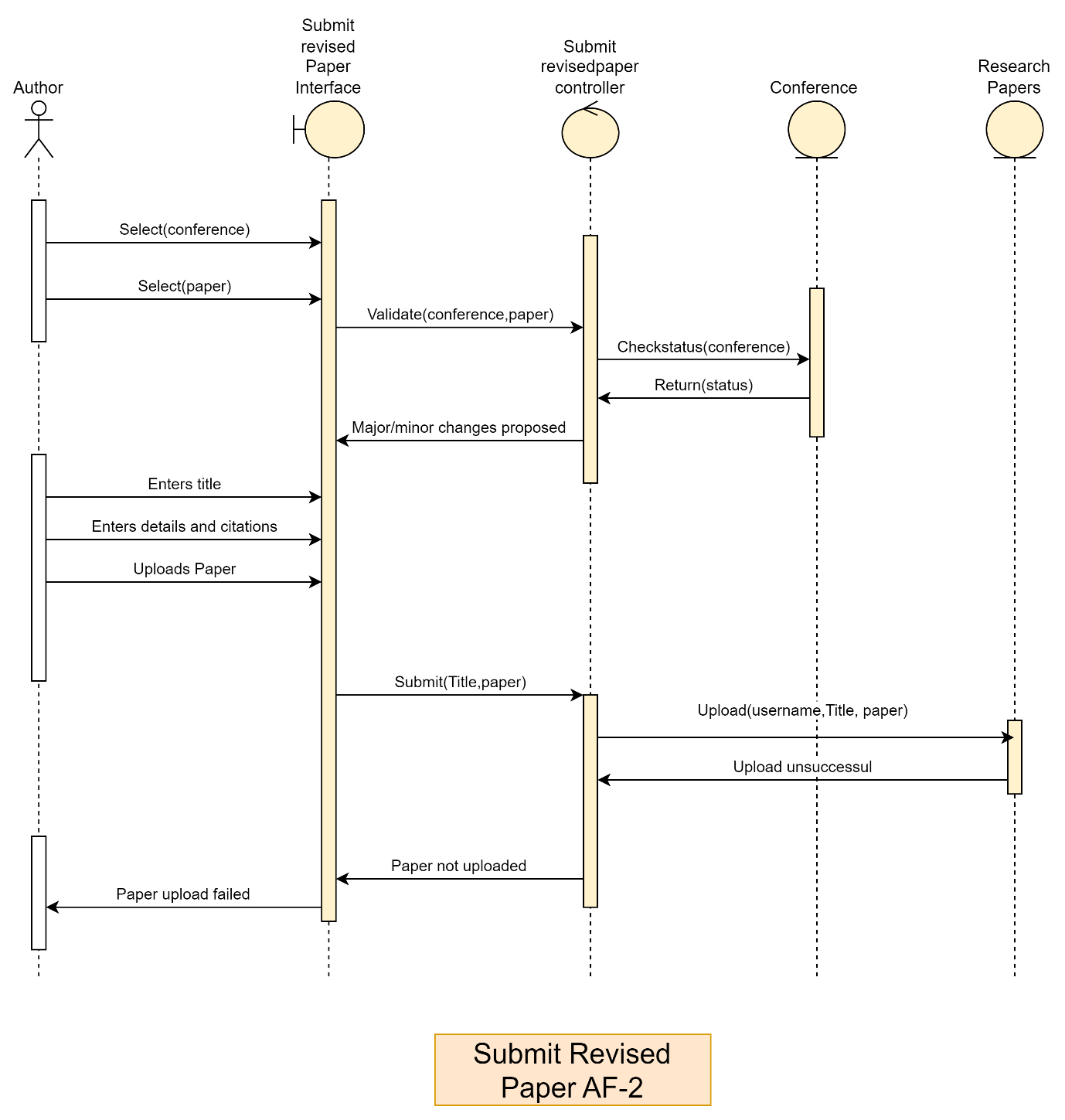


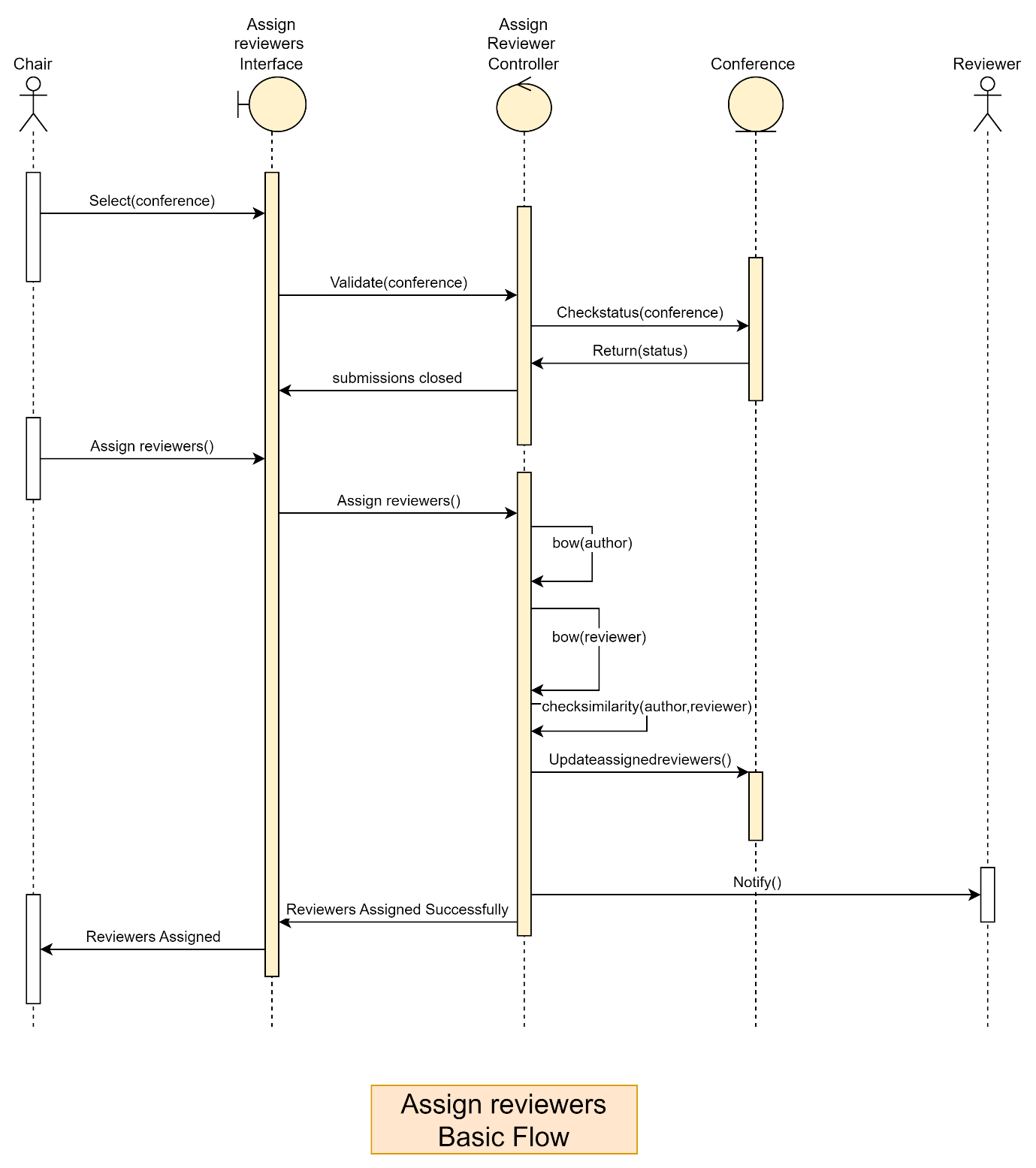


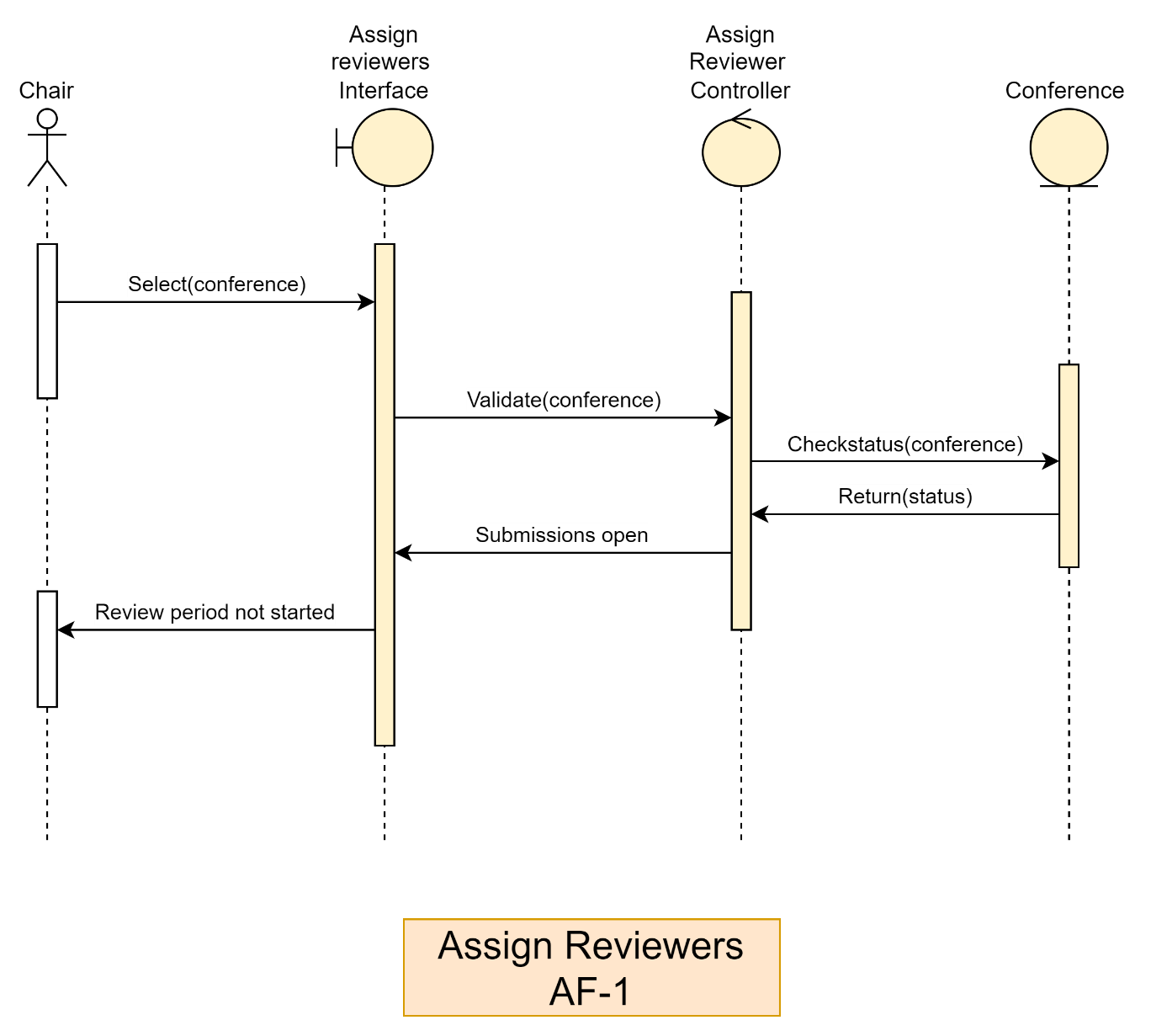












**8. TEST CASES**

1. **Registration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks** |
| TC001 | Successful Registration | Valid personal info | Valid academic info | Account successfully registered | None |
| TC002 | Passwords do not match | Password1 | Password2 | Prompt to re-enter password | Test invalid pwd |
| TC003 | Username already exists | Existing username | Valid password | Prompt to choose another username | None |
| TC004 | Missing mandatory fields | Blank fields | - | Prompt to complete mandatory fields | None |
| TC005 | Invalid academic information | Invalid data format | - | Prompt to enter valid academic information | None |
| TC006 | Successful Registration with edge values | Edge case data | Valid academic info | Account successfully registered | Boundary testing |
| TC007 | Invalid phone number format | Invalid phone | Valid personal info | Prompt to enter a valid phone number | Validate input |
| TC008 | Role not selected | No role selected | - | Prompt to select a role | Test edge case |
| TC009 | Duplicate email address | Existing email | Valid personal info | Prompt to use a different email | None |
| TC010 | Invalid graduation year | Invalid year | Valid personal info | Prompt to enter a valid year | None |
| TC011 | System timeout during registration | Valid input | - | Prompt to retry the operation | Handle failures |
| TC012 | Special characters in name field | Invalid name | - | Prompt to enter a valid name without special characters | Validate input |
| TC013 | Maximum length exceeded for username | Long username | Valid password | Prompt to enter a shorter username | None |
| TC014 | Registration with minimum input | Minimum valid data | - | Account successfully registered | Test edge case |
| TC015 | Invalid specialization field | Invalid specialization | - | Prompt to enter a valid specialization | Validate input |

1. **Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks** |
| TC001 | Successful Login | Valid username | Valid password | User is logged in and redirected to dashboard | None |
| TC002 | Incorrect Password | Valid username | Invalid password | Prompt to re-enter password | Test invalid pwd |
| TC003 | Incorrect Username | Invalid username | Valid password | Prompt to re-enter username | None |
| TC004 | Both Username and Password Incorrect | Invalid username | Invalid password | Prompt to re-enter credentials | None |
| TC005 | Missing Username | Blank username | Valid password | Prompt to enter username | None |
| TC006 | Missing Password | Valid username | Blank password | Prompt to enter password | None |
| TC007 | System timeout during login | Valid username | Valid password | Prompt to retry the operation | Handle failures |
| TC008 | Locked account due to multiple failures | Valid username | Invalid password | Account locked after multiple failed attempts | None |
| TC009 | Login with special characters in username | Special chars in username | Valid password | Prompt to enter valid username | Validate input |
| TC010 | Case-sensitive password check | Valid username | Incorrect case password | Prompt to re-enter password | Ensure case-check |
| TC011 | Minimum input length validation | Short username | Short password | Prompt to enter valid credentials | Validate input |
| TC012 | Maximum username length exceeded | Very long username | Valid password | Prompt to enter a shorter username | Boundary testing |
| TC013 | Maximum password length exceeded | Valid username | Very long password | Prompt to enter a shorter password | Boundary testing |
| TC014 | Login with unregistered username | Non-existent username | Valid password | Prompt that username is not registered | None |
| TC015 | Login with expired account | Valid username | Valid password | Prompt that account has expired | Handle failures |
| TC016 | Login after account reactivation | Valid reactivated username | Valid password | User is logged in successfully | Test edge case |
| TC017 | SQL injection attempt in username | SQL payload | Valid password | Prompt to enter valid credentials | Security check |
| TC018 | SQL injection attempt in password | Valid username | SQL payload | Prompt to enter valid credentials | Security check |
| TC019 | Login with session already active | Valid username | Valid password | User is warned about an active session | Session handling |
| TC020 | Login with username containing spaces | Username with spaces | Valid password | Prompt to enter a valid username | Validate input |

1. **Submit Paper**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks** |
| TC001 | Successful Submission | Valid credentials | Valid paper file | Submission registered successfully | None |
| TC002 | Conference submissions not open | Conference state = closed | Valid credentials | Prompt that submissions are not open | Verify conference state |
| TC003 | Invalid file type uploaded | Valid credentials | Invalid file format | Prompt to upload a valid file format | Validate file upload |
| TC004 | Missing credentials | Blank credentials | Valid paper file | Prompt to enter credentials | None |
| TC005 | Missing paper file | Valid credentials | No file uploaded | Prompt to upload a file | None |
| TC006 | File size exceeds limit | Valid credentials | File > allowed size limit | Prompt to upload a smaller file | Validate file size |
| TC007 | System timeout during file upload | Valid credentials | Valid paper file | Prompt to retry the operation | Handle failures |
| TC008 | Duplicate paper submission | Valid credentials | File already submitted | Prompt that paper is already submitted | Check for duplicates |
| TC009 | Unauthorized user attempts submission | Unauthorized user credentials | Valid paper file | Prompt unauthorized access | None |
| TC010 | Conference state: Review period started | Conference state = review | Valid credentials | Prompt that submissions are closed | Verify conference state |
| TC011 | Paper with special characters in metadata | Valid credentials | File with special metadata | Submission registered successfully | Test edge case |
| TC012 | Concurrent submissions from the same user | Valid credentials | Two simultaneous uploads | Only one submission is registered | Handle concurrency |
| TC013 | Invalid characters in keywords field | Invalid keywords | Valid paper file | Prompt to enter valid keywords | Validate input |
| TC014 | Missing mandatory fields in submission form | Blank form fields | Valid paper file | Prompt to fill mandatory fields | None |
| TC015 | Submission during system maintenance | Valid credentials | Valid paper file | Prompt that system is under maintenance | System downtime case |
| TC016 | Submission with expired session | Expired user session | Valid paper file | Prompt to re-login | Test session handling |
| TC017 | Submission with invalid file name | Valid credentials | File with invalid name | Prompt to rename the file | Validate file naming |
| TC018 | Simultaneous submissions from different users | Two valid user credentials | Two valid paper files | Both submissions registered successfully | Test concurrency |
| TC019 | Paper re-submission with changes | Valid credentials | Updated paper file | Submission updated successfully | None |
| TC020 | SQL injection in metadata fields | Malicious metadata | Valid paper file | Prompt to enter valid data | Security validation |
| TC021 | Paper upload interrupted due to network error | Valid credentials | Paper upload interrupted | Prompt to retry upload | Handle network errors |
| TC022 | Submission with plagiarized content | Valid credentials | Plagiarized paper file | Prompt about detected plagiarism | Requires plagiarism check |
| TC023 | Submission with unsupported file extension | Valid credentials | File with unsupported extension | Prompt to use supported formats | Validate file types |
| TC024 | Submission with uncompressed large file | Valid credentials | Uncompressed paper file | Prompt to compress file | Test file handling |
| TC025 | Metadata contains HTML/JavaScript code | Metadata with scripts | Valid paper file | Prompt to enter valid text | Security validation |

1. **Review Paper**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected Output** | **Remarks** |
| TC01 | Reviewer logs in and accesses the assigned papers page during the review period. | Reviewer credentials | Conference timeline state: "Review period started" | Reviewer successfully logs in and views assigned papers. | Ensure valid credentials. |
| TC02 | Reviewer tries to access assigned papers page before the review period starts. | Reviewer credentials | Conference timeline state: "Submission are available" | System denies access and displays a message: "Review period has not started yet." | Verify timeline state checking. |
| TC03 | Reviewer accesses the system with invalid credentials. | Invalid reviewer credentials | - | System denies login and shows an error message: "Invalid username or password." | Validate error handling. |
| TC04 | Reviewer submits feedback for a paper successfully. | Review feedback content | Paper ID | System saves the review and confirms: "Review submitted successfully." | Test for valid review content. |
| TC05 | Reviewer tries to submit feedback with incomplete or empty content. | Empty feedback content | Paper ID | System prevents submission and displays an error message: "Review content cannot be empty." | Ensure input validation. |
| TC06 | Reviewer logs out of the system after viewing assigned papers. | Reviewer credentials | - | System logs out the reviewer and ends the session successfully. | Confirm session closure. |
| TC07 | Reviewer accesses the assigned papers page after the review period ends. | Reviewer credentials | Conference timeline state: "Review period ended" | System denies access and displays a message: "Review period has ended." | Ensure end-state enforcement. |
| TC08 | System verifies conference state when accessing assigned papers page. | Reviewer credentials | Conference timeline state: Any (valid states) | System allows or denies access based on the conference state. | Test for timeline state checks. |
| TC09 | Reviewer submits a review for a paper not assigned to them. | Review feedback content | Unassigned Paper ID | System denies submission and displays an error message: "You are not authorized to review this paper." | Ensure access controls. |
| TC10 | System maintains consistent database state after review submission. | Valid review feedback | Assigned Paper ID | Conference and paper databases reflect accurate updates after review submission. | Verify database consistency. |
| TC11 | Reviewer attempts to access the system while the conference is in "Submission are available" state. | Reviewer credentials | Conference timeline state: "Submission are available" | System denies access and displays a message: "Review period has not started yet." | Verify proper state transition logic. |
| TC12 | Reviewer attempts to submit feedback with invalid content (e.g., non-text content). | Non-text review content | Paper ID | System rejects submission and displays an error message: "Invalid content format." | Test for content validation. |
| TC13 | Reviewer attempts to submit feedback after the review period has ended. | Review feedback content | Paper ID | System denies submission and displays a message: "Review period has ended." | Verify post-period feedback restriction. |
| TC14 | Reviewer views the paper but decides not to submit any feedback. | No feedback content | Paper ID | Reviewer can exit without submitting, and the paper status remains unchanged. | Confirm proper exit behavior. |
| TC15 | Reviewer attempts to submit a review while already having submitted feedback for the same paper. | Review feedback content | Paper ID | System displays a message: "You have already submitted a review for this paper." | Ensure review duplication prevention. |
| TC16 | Reviewer tries to view the paper and the system detects a conflict (e.g., paper no longer assigned). | Reviewer credentials | Invalid Paper ID | System displays a message: "Paper is no longer assigned to you." | Validate assignment tracking. |
| TC17 | Reviewer submits a review with an attachment (if allowed by the system). | Review feedback content with attachment | Paper ID | System accepts review submission with attachment and confirms: "Review submitted successfully." | Ensure attachment handling. |
| TC18 | Reviewer tries to submit feedback with a submission that exceeds the character limit (if specified). | Excessive review content | Paper ID | System prevents submission and displays an error message: "Review content exceeds the character limit." | Test for input size validation. |
| TC19 | Reviewer accesses the system after a session timeout. | Reviewer credentials | - | System prompts for login again and allows the reviewer to log back in. | Test for session timeout handling. |
| TC20 | Reviewer submits feedback after revising it multiple times. | Revised review feedback | Paper ID | System saves the most recent feedback version and confirms: "Review updated successfully." | Verify feedback version handling. |

1. **Check Status/Feedback**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks (if any)** |
| TC001 | Verify login functionality before accessing status/feedback | Valid username | Valid password | Author successfully logs in | Ensure credentials are valid and user exists. |
| TC002 | Verify access without logging in | N/A | N/A | System restricts access to the profile page and prompts for login | Unauthorized access should be blocked. |
| TC003 | Check access to submissions tab | Logged-in author | N/A | Submissions tab is accessible | Ensure no errors in navigation to the tab. |
| TC004 | Verify display of submissions information for an author | Valid logged-in user | N/A | System displays the list of all submissions by the author | Ensure submissions displayed match the logged-in author. |
| TC005 | Verify feedback display for a selected submission | Selected submission | N/A | System displays reviewer feedback corresponding to the selected submission | Feedback should be clearly visible to the author. |
| TC006 | Verify behavior for an author with no submissions | Logged-in author | N/A | System displays a message indicating no submissions are available | Should gracefully handle scenarios where no data exists. |
| TC007 | Verify proper status display for submitted papers | Logged-in author | N/A | System displays the current status (e.g., *Under Review*, *Accepted*, *Rejected*) for all submissions | Status values should match data in the database. |
| TC008 | Verify system behavior on session timeout | Active session | N/A | System logs out the author and prompts for login again when accessing the profile page | Ensure session timeout is handled securely. |
| TC009 | Verify resuming flow after system halt | Valid username | Valid password | Author logs in again, navigates to the profile page, and accesses submissions tab without errors | System should maintain data integrity and availability upon resuming operations. |
| TC010 | Verify system behavior with invalid login credentials | Invalid username | Invalid password | System displays an error message and denies access | Ensure error messages are clear and informative. |
| TC011 | Verify response when feedback is not available for a submission | Selected submission | N/A | System displays a message indicating no feedback has been provided yet | Feedback field should not be empty or confusing. |
| TC012 | Verify exit functionality from the system | Logged-in author | N/A | System logs out the author successfully | Should ensure all sessions are terminated securely. |
| TC013 | Check system behavior when database is unavailable | Logged-in author | N/A | System displays an error message and fails gracefully | Ensure database errors are communicated effectively without exposing sensitive info. |
| TC014 | Verify display for authors with multiple submissions | Logged-in author | N/A | System displays all submissions with corresponding statuses and feedback | UI should handle multiple records without issues. |
| TC015 | Verify pagination or scrolling for large lists of submissions | Logged-in author | N/A | System provides pagination or infinite scrolling for submissions list | Important for usability with a large dataset. |

1. **Logout**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks (if any)** |
| TC001 | Verify successful logout functionality | Logged-in user | N/A | System terminates the session, clears local data, and redirects to the login page with a success message | Verify the user sees a "logged out successfully" message. |
| TC002 | Verify logout functionality for a chair | Logged-in chair | N/A | System logs out the chair, clears session data, and redirects to the login page | Ensure no role-specific session data is left behind. |
| TC003 | Verify logout functionality for an author | Logged-in author | N/A | System logs out the author, clears session data, and redirects to the login page | Ensure data security is maintained post-logout. |
| TC004 | Verify logout functionality for a reviewer | Logged-in reviewer | N/A | System logs out the reviewer, clears session data, and redirects to the login page | Role-specific workflows should behave consistently. |
| TC005 | Verify logout button visibility | Logged-in user | N/A | Logout button/link is visible and functional | Ensure button is accessible on all screens. |
| TC006 | Verify redirection to the login page | Logout initiated | N/A | User is redirected to the login page after logging out | Check if redirection is consistent across roles. |
| TC007 | Verify that session data is cleared | Logout initiated | N/A | All session data (cookies, tokens, etc.) are deleted | No residual session data should persist post-logout. |
| TC008 | Verify logout on session timeout | Session timeout | N/A | System automatically logs out the user and redirects to the login page | Validate proper handling of session expiry. |
| TC009 | Verify logout functionality when the system is under heavy load | Logged-in user | High load | Logout process completes successfully, and user is redirected to the login page | Ensure robustness under heavy traffic. |
| TC010 | Verify logout when accessing from a shared/public device | Logged-in user | Shared device | System logs out the user and clears all sensitive data | Ensure no traces of the user’s session remain accessible. |
| TC011 | Verify logout functionality for an inactive user | Inactive session | N/A | System logs out the user automatically and redirects to the login page | Check auto-logout for idle sessions. |
| TC012 | Verify logout functionality with multiple browser tabs open | Logout initiated | N/A | Logout from one tab logs out the user from all open tabs | Test session synchronization across tabs. |
| TC013 | Verify confirmation message post-logout | Logout initiated | N/A | System displays a confirmation message like "You have been logged out successfully" | Ensure clarity in confirmation messaging. |
| TC014 | Verify logout behavior during active operation (e.g., submission) | Logout initiated | N/A | System terminates session and prompts the user to re-authenticate to resume operations | Check how interrupted workflows are handled. |
| TC015 | Verify behavior when accessing restricted pages post-logout | Logout initiated | N/A | System restricts access to restricted pages and redirects to the login page | Validate that unauthorized access is blocked. |
| TC016 | Verify logout with server-side session termination | Logout initiated | N/A | System terminates the server-side session and clears local session data | Ensure sessions are invalidated server-side. |
| TC017 | Verify logout link accessibility via keyboard navigation | Logged-in user | Keyboard | Logout link/button is accessible via keyboard shortcuts or tab navigation | Ensure compliance with accessibility standards. |
| TC018 | Verify logout functionality for different browsers | Logout initiated | Browser type | Logout functionality works consistently across multiple browsers (e.g., Chrome, Firefox, Safari) | Test cross-browser compatibility. |
| TC019 | Verify logout after password change | Password updated | N/A | User remains logged in after password change but can logout and re-login with the new password | Test integration with password update flows. |
| TC020 | Verify logout behavior when the backend server is down | Logout initiated | Server down | System gracefully handles server issues and informs the user about the error | Ensure the user receives an appropriate message if logout cannot be processed. |

1. **Submit feedback**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks (if any)** |
| TC001 | Verify navigation to "Submit Feedback" section | Logged-in reviewer | N/A | Reviewer successfully navigates to the "Submit Feedback" section | Reviewer must have proper access rights. |
| TC002 | Verify system checks conference timeline | Conference status | N/A | System allows access only if the "Review period has started" | Conference status must match the required state. |
| TC003 | Verify behavior when reviews are not opened yet | Conference status | N/A | System displays a message and ends the use case | Graceful handling when timeline conditions are unmet. |
| TC004 | Verify display of the list of submitted papers | Logged-in reviewer | N/A | System displays all papers assigned to the reviewer | Ensure only authorized papers are displayed. |
| TC005 | Verify paper selection functionality | Selected paper ID | N/A | System displays the feedback submission form for the selected paper | Ensure accurate data is shown for the selected paper. |
| TC006 | Verify input of feedback in the form | New feedback text | New status (dropdown) | System accepts feedback and validates the new status | Ensure all form fields are functioning correctly. |
| TC007 | Verify validation of required fields | Empty feedback | Selected status | System prompts the reviewer to fill in the feedback field | Required fields must be enforced. |
| TC008 | Verify update with an invalid status | New feedback | Invalid status | System rejects invalid status and prompts the reviewer | System must validate against predefined status options. |
| TC009 | Verify successful feedback submission | Valid feedback | Valid status | System updates the database and displays a success message | Ensure smooth operation for valid inputs. |
| TC010 | Verify database update for submitted feedback | Valid feedback | Valid status | Database is updated with the new feedback and status | Check for consistent and accurate data updates. |
| TC011 | Verify notification sent to the associated author | Valid feedback submission | Author ID | System sends an email or in-app notification to the author about the updated feedback | Ensure notifications are sent without delay. |
| TC012 | Verify system behavior for unauthorized reviewers | Unauthorized reviewer login | N/A | System denies access to the "Submit Feedback" section | Access control must be strictly enforced. |
| TC013 | Verify logout during feedback submission process | Logged-in reviewer logs out | N/A | System terminates the session and prevents incomplete submissions | Ensure proper session handling and security. |
| TC014 | Verify feedback submission for multiple papers | Feedback for paper 1 | Feedback for paper 2 | System processes feedback for all selected papers without errors | Test bulk operations or repeated submissions. |
| TC015 | Verify behavior when the server is down during submission | Valid feedback | Valid status | System displays an error message and does not update the database | Ensure no partial or inconsistent updates occur. |
| TC016 | Verify display of feedback submission success message | Valid feedback | Valid status | System shows a success message confirming submission | Feedback confirmation improves user experience. |
| TC017 | Verify notification failure scenario | Valid feedback submission | Invalid author contact | System logs the notification failure and continues without blocking the reviewer | Notification system failures should not impact feedback submission. |
| TC018 | Verify form reset functionality | Feedback entered | Reset button clicked | System clears all entered data in the form | Useful for corrections before final submission. |
| TC019 | Verify feedback submission with special characters in the text | Special character feedback | Valid status | System accepts the feedback without any errors | Ensure proper handling of text input encoding. |
| TC020 | Verify feedback submission on a paper that already has feedback | Feedback entered | Existing feedback | System replaces or appends the feedback based on the conference rules | Ensure proper rules are followed for updates to existing feedback. |

1. **Submit revised paper**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks (if any)** |
| TC001 | Verify navigation to "Submit Revised Paper" section | Logged-in author | N/A | Author successfully navigates to the "Submit Revised Paper" section | Author must be logged in and authorized. |
| TC002 | Verify conference timeline check | Conference status | N/A | System allows access only if the "Review period started" state is active | Conference status must meet the required condition. |
| TC003 | Verify behavior when the review period has not started | Conference status | N/A | System ends the use case with an appropriate message | Graceful handling when conditions are unmet. |
| TC004 | Verify display of submission form | Logged-in author | N/A | System displays the form with necessary fields (e.g., Name, Keywords, File Upload) | Ensure the form is user-friendly and complete. |
| TC005 | Verify entry of required fields | Name | Keywords | System accepts and validates the input for Name and Keywords | Ensure validation of required fields. |
| TC006 | Verify file upload functionality | Valid file | N/A | System successfully uploads the file and validates the file format | Check support for allowed file formats (e.g., PDF). |
| TC007 | Verify file upload with an unsupported file format | Unsupported file format | N/A | System rejects the file and prompts the user to upload a supported format | Ensure proper error messaging for invalid formats. |
| TC008 | Verify system response when no file is uploaded | Empty file field | N/A | System prompts the user to upload a file before proceeding | Validate mandatory file upload requirement. |
| TC009 | Verify submission of valid revised paper | Valid form data | Valid file | System accepts the revised paper and displays a success message | Ensure smooth operation for valid inputs. |
| TC010 | Verify database update after successful submission | Valid revised paper | Author ID | System updates the database with the new paper version and metadata | Ensure consistency and integrity of data updates. |
| TC011 | Verify system behavior when a paper without requested changes is uploaded | Invalid paper ID | N/A | System rejects the upload and informs the author that changes were not requested | Ensure system enforces rules for revisions. |
| TC012 | Verify behavior when the system is under heavy load | Valid revised paper | N/A | System processes the submission without errors | Ensure robustness under heavy traffic. |
| TC013 | Verify logout during the submission process | Author logs out | N/A | System terminates the session and prevents incomplete submissions | Ensure proper session handling and security. |
| TC014 | Verify duplicate file uploads | Same file uploaded twice | N/A | System replaces the existing file or notifies the user appropriately | Ensure user awareness of duplicate actions. |
| TC015 | Verify confirmation message after successful submission | Valid revised paper | N/A | System displays a confirmation message like "Your revised paper has been submitted successfully" | Confirmation improves user experience. |
| TC016 | Verify behavior when server-side error occurs | Valid revised paper | N/A | System displays an error message and does not update the database | Ensure graceful failure handling. |
| TC017 | Verify maximum file size restriction | File exceeding size limit | N/A | System rejects the file and prompts the user to upload a smaller file | Ensure system enforces file size limits. |
| TC018 | Verify handling of special characters in form fields | Special characters in Name | Keywords | System accepts and validates the fields correctly | Validate input encoding and sanitization. |
| TC019 | Verify notification to authors after successful submission | Valid revised paper | Author ID | System sends a notification (email or in-app) to the author about the successful submission | Ensure notifications are sent without delay. |
| TC020 | Verify system behavior when the conference database is unavailable | Valid revised paper | N/A | System displays an error message and ensures no partial or inconsistent updates | Database issues should not compromise system stability. |

1. **Assign reviewers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks (if any)** |
| TC001 | Verify navigation to "Assign Reviewers" section | Logged-in chair | N/A | Program chair successfully navigates to the "Assign Reviewers" section | Chair must have the necessary access rights. |
| TC002 | Verify conference timeline check | Conference state | N/A | System allows access only if the "Review period has started" | System must restrict actions based on conference state. |
| TC003 | Verify behavior when the review period has not started | Conference state | N/A | System ends the use case with an appropriate message | Proper error handling for unmet preconditions. |
| TC004 | Verify start of assignment scenario | Valid conference chair login | N/A | System starts the assignment process and displays relevant details | Ensure the interface provides necessary details for assignment. |
| TC005 | Verify bag of words computation for reviewers | Reviewers' metadata | Papers metadata | System computes bag of words for reviewers and moves to the next step | Ensure accurate computation and validation. |
| TC006 | Verify behavior when bag of words for reviewers is not precomputed | Reviewer metadata | N/A | System computes the bag of words for reviewers and continues | Test system resilience and fallback mechanisms. |
| TC007 | Verify bag of words computation for authors | Authors' metadata | Papers metadata | System computes bag of words for authors and moves to the next step | Validate accuracy of author-papers mapping. |
| TC008 | Verify behavior when bag of words for authors is not precomputed | Author metadata | N/A | System computes the bag of words for authors and continues | Ensure fallback for missing precomputed data. |
| TC009 | Verify similarity check between papers | Bag of words for reviewers | Bag of words for authors | System calculates similarity scores and identifies suitable reviewers | Verify precision in similarity computation. |
| TC010 | Verify conflict of interest detection | Conflict metadata | Reviewer metadata | System identifies potential conflicts and flags them | Ensure conflicts are identified accurately. |
| TC011 | Verify behavior when conflicts of interest are detected | Conflict metadata | N/A | System computes bag of words again to resolve conflicts and continues | Conflict resolution mechanisms should be robust. |
| TC012 | Verify paper assignment process | Valid paper ID | Reviewer ID | System assigns the paper to the reviewer and updates the database | Ensure consistent and accurate assignment updates. |
| TC013 | Verify unassigned papers detection after the assignment process | Papers metadata | Reviewer assignments | System identifies unassigned papers and displays them to the program chair | Test system accountability for complete assignments. |
| TC014 | Verify system behavior when all reviewers have maximum workload | Reviewer metadata | Papers metadata | System balances workloads and assigns papers fairly | Ensure no reviewer exceeds workload limits. |
| TC015 | Verify error message when no suitable reviewers are available | Papers metadata | Reviewer metadata | System notifies the program chair that no suitable reviewers are available | Ensure system provides actionable feedback. |
| TC016 | Verify database consistency after successful assignment | Valid assignments | Database state | Database updates are consistent with the new reviewer assignments | Test for data consistency and integrity. |
| TC017 | Verify system behavior when the assignment process is interrupted | Mid-assignment | N/A | System saves progress and allows resumption without data loss | Ensure proper handling of interrupted processes. |
| TC018 | Verify system response to invalid reviewer or paper data | Invalid reviewer ID | Invalid paper ID | System rejects invalid data and prompts the program chair to provide valid inputs | Validate input data thoroughly. |
| TC019 | Verify behavior when the system fails to save assignments | Valid assignments | Database unavailable | System displays an error message and ensures no partial updates occur | Test system resilience to backend failures. |
| TC020 | Verify assignment completion confirmation | All papers assigned | N/A | System displays a confirmation message indicating that all papers have been successfully assigned | Confirmation improves user confidence and system transparency. |

***8.10 Manage Conference***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario & Description** | **Input 1** | **Input 2** | **Expected O/P** | **Remarks (if any)** |
| TC001 | Verify navigation to "Manage Conference" section | Logged-in chair | N/A | Program chair successfully navigates to the "Manage Conference" section | Chair must have the necessary permissions. |
| TC002 | Verify option to add or view conferences | Manage Conference action | N/A | System displays options: "Add conference" and "View conference" | Ensure intuitive and accessible UI for selection. |
| TC003 | Verify Add Conference sub-flow is triggered | "Add conference" | N/A | System redirects to the conference submission form | Ensure the transition is smooth and error-free. |
| TC004 | Verify View Conference sub-flow is triggered | "View conference" | N/A | System displays a list of all ongoing conferences | Validate correct data retrieval and display. |
| TC005 | Verify conference submission form is displayed | "Add conference" | N/A | System displays a form for entering conference details | Ensure form layout is user-friendly. |
| TC006 | Verify required fields in conference submission form | Conference Name | Deadlines | System accepts valid input for all required fields | Ensure mandatory fields are validated. |
| TC007 | Verify invalid data in conference submission form | Invalid deadlines | Conference Name | System rejects invalid data and prompts the user to correct errors | Validate data input thoroughly. |
| TC008 | Verify system behavior when entering duplicate conference data | Duplicate Conference Name | Valid Deadlines | System alerts the user about duplicate entries and prevents submission | Ensure unique conference entries. |
| TC009 | Verify successful registration of new conference | Valid Name and Deadlines | N/A | System saves the new conference data and displays a success message | Confirm database update is consistent and accurate. |
| TC010 | Verify database consistency after failed save | Valid conference data | Database unavailable | System ensures no partial updates occur in the database | Test robustness in case of backend failures. |
| TC011 | Verify system response when no ongoing conferences are available | "View conference" | N/A | System displays a message indicating no ongoing conferences | Validate proper feedback for empty results. |
| TC012 | Verify display of conference papers and their statuses | Selected conference | N/A | System displays the list of submitted papers with their statuses | Ensure accurate retrieval of paper metadata. |
| TC013 | Verify proper handling of rejected papers | Selected conference | Rejected papers | System clearly indicates the rejection status of papers | Ensure visibility and clarity of paper statuses. |
| TC014 | Verify proper handling of papers with proposed changes | Selected conference | Papers with changes | System clearly indicates the "Changes proposed" status | Test proper categorization and display of papers. |
| TC015 | Verify session management during conference addition | Logged-out chair | N/A | System terminates the session and requires re-login | Ensure security and proper session handling. |
| TC016 | Verify system behavior during unexpected halts | Mid-add conference | N/A | System saves progress and allows resumption after re-login | Ensure data loss is avoided during interruptions. |
| TC017 | Verify notifications after successful conference addition | Valid conference data | N/A | System sends notification (email or in-app) confirming the addition of a new conference | Improve user experience with timely notifications. |
| TC018 | Verify handling of simultaneous conference additions | Multiple chairs adding | Valid data | System processes each addition independently and ensures no conflicts | Test for concurrency management. |
| TC019 | Verify system behavior when invalid conference selection is made | Invalid conference ID | N/A | System alerts the user and prevents viewing of invalid conference details | Ensure robustness to handle invalid inputs. |
| TC020 | Verify system's ability to handle large conference data sets | Many conferences | Papers per conference | System displays data efficiently without lag or crashes | Test scalability and performance under heavy load. |