PES UNIVERSITY

Department of Computer Science

and Engineering

UE22CS341A: Software

Engineering

Software Requirements Specification for Conference Paper Management System

Name : Vedant Varma
SRN : PES1UG22CS681

Name : Vivek M
SRN : PES1UG22CS707

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Synopsis :

This document outlines the requirements for the Conference Paper Management System, designed to facilitate interactions between authors, reviewers, and readers. The system allows authors to submit and manage their papers, reviewers to provide feedback, and readers to search for and access published research. It follows the AGILE model, emphasizing flexibility in development.

Key Sections:

Purpose & Scope: The system provides a platform for academic paper submission, review, and discovery. It streamlines the publication process and ensures efficient management of academic papers and reviews.

User Classes: The system serves three primary user groups—Authors, Reviewers, and Readers—with tailored features for each, such as paper submission and profile management for authors, paper review tools for reviewers, and advanced search functionalities for readers.

Product Functions:

Author and Reviewer registration and profile management
Paper submission and status tracking
Reviewer management with paper assignments and feedback submission
Search and retrieval of research papers based on various filters
Journals and conference management, including paper submission and viewing metrics
Interfaces:

Web-based user interface with a responsive design
Backend framework (Flask) and frontend using React for dynamic content delivery
Database (MySQL) for storing user data, paper submissions, and reviews
Secure communication through HTTPS and encryption
Functional Requirements: Detailed requirements for user registration, paper submission, review management, and search functionality. Each user action prompts specific system responses, ensuring a seamless user experience.

Non-Functional Requirements:

Performance: Supports up to 1,000 concurrent users, with quick data retrieval and paper downloads.
Security: Ensures encryption of user data and implements multi-factor authentication.
Software Quality: Focuses on adaptability, availability, correctness, portability, reliability, and usability.

1. Introduction

1.1 Purpose

This document outlines the requirements for the Conference Paper Management System. The system will serve as an intermediary application, facilitating interaction between authors, reviewers, and readers. The system allows authors to submit papers, reviewers to manage reviews, and readers to browse and search for published papers. This document is intended for both the stakeholders and the developers of the system.

The model aims to use **AGILE** model due to it's dynamic nature which helps in our development.

1.2 Intended Audience

The document is intended for developers, project managers, marketing staff, testers, documentation writers and users who comprise of Readers, Authors and Reviewers.

1.3 Scope

The system is a comprehensive platform for managing academic publications, designed to streamline interactions among authors, reviewers, and readers. Authors can upload their manuscripts, provide detailed metadata, and manage revisions through the system. It supports the entire submission process, including tracking paper updates and handling supplementary materials. Reviewers are integral to the peer review process, using the system to provide structured feedback, assess paper quality, and communicate with authors. Readers benefit from advanced search and browsing features, allowing them to discover and access published content easily. The system maintains extensive databases for papers, authors, and review data, ensuring efficient retrieval and secure management of sensitive information.

1.4 References

- IEEE Xplore (https://ieeexplore.ieee.org/Xplore/home.jsp)
- EasyChair (https://easychair.org/)
- IEEE Standard for Software Requirements Specifications (IEEE Std 830 1998)

2. Overall Description

2.1 Product Perspective

The system is an online platform accessible via web browsers. It serves as a database housing published research papers, thereby allowing users to access and read papers of their interest. It provides a channel for authors to submit paper drafts to diverse conferences. Reviewers can review through papers and reach out to the authors with potential feedback. It is designed to cater to authors, reviewers, and readers involved in academic research and publications.

2.2 Product Functions

- Author and Reviewer registration and profile management
- Paper submission and status tracking
- Search and retrieval of papers filtered on various specific constraints
- Reviewers can view assigned papers, submit reviews, and track their review history.
- User interaction and analytics (feedback)

2.3 User Classes and Characteristics

- **Authors**: Researchers who register to upload, manage, and track their papers.
- **Reviewers**: Academics who review submitted papers and provide feedback.
- **Readers**: General users who search for and read available papers and articles.

2.4 Operating Environment

- Runs on web browsers (Chrome, Firefox, Safari) with a responsive UI.
- Server environment with a database and web server for backend processing.
- No hardware required.

2.5 Design and Implementation Constraints

- Compliance with data protection laws (e.g., GDPR)
- Scalability to handle multiple conferences and large datasets.
- Secure data transmission

2.6 Assumptions and Dependencies

- Authors and reviewers have internet access.
- The operating environment (e.g, MySQL database) will remain stable and fully supported for the duration of the project
- Regular system maintenance and backups will be performed.
- Assuming that no content published is plagiarized.

3. External Interface Requirements

3.1 User Interfaces

- Web-based interface with responsive design for desktop access.
- Dashboard for authors to manage papers and profiles.
- Dashboard to view diverse conferences available and respective papers published
- Review portal for reviewers to access papers.
- Search and filter tools for readers.

3.2 Software Interfaces

This section describes the connections between the Conference Paper Management System and other specific software components, including databases, operating systems, tools, libraries, and integrated commercial components. It outlines the data flow, communication protocols, and services required for seamless integration and operation.

3.2.1 Database Interfaces

- **MySQL (Version 7+)**:
  - **Purpose**: Used as the primary relational database management system for storing structured data, including user profiles, paper submissions, review details, and journal/conference information.
  - **Data Flow**:
    - **Incoming Data**: User registration details, paper metadata, submission data, and review feedback.
    - **Outgoing Data**: User profiles, search results, submission statuses, and reports.
  - **Communication**: The system will communicate with MySQL using SQL queries over a secure connection (e.g., SSL).

3.2.2 Application Frameworks and Libraries

- **Backend Framework (Flask)**
  - **Purpose**: Provides the application framework for managing web requests , handling business logic, and interfacing with the database.
  - **Communication**: Uses HTTP/HTTPS for RESTful API communication between the client and server, handling user requests like submission uploads and status checks.
  - **Data Flow**:
    - **Incoming Data**: User inputs from forms (e.g., login credentials, paper details).
    - **Outgoing Data**: JSON responses for client-side rendering (e.g., search results, user notifications).
- **Frontend Framework (React 17+ or HTML,CSS,JS)**:
  - **Purpose**: Handles the user interface, providing a dynamic, responsive experience across devices.
  - **Communication**: Uses RESTful APIs to communicate with the backend, fetching and displaying data dynamically.
  - **Data Flow**:
    - **Incoming Data**: API responses containing user data, paper details, or search results.
    - **Outgoing Data**: User actions, such as form submissions or search queries.

3.3 Communication Interfaces

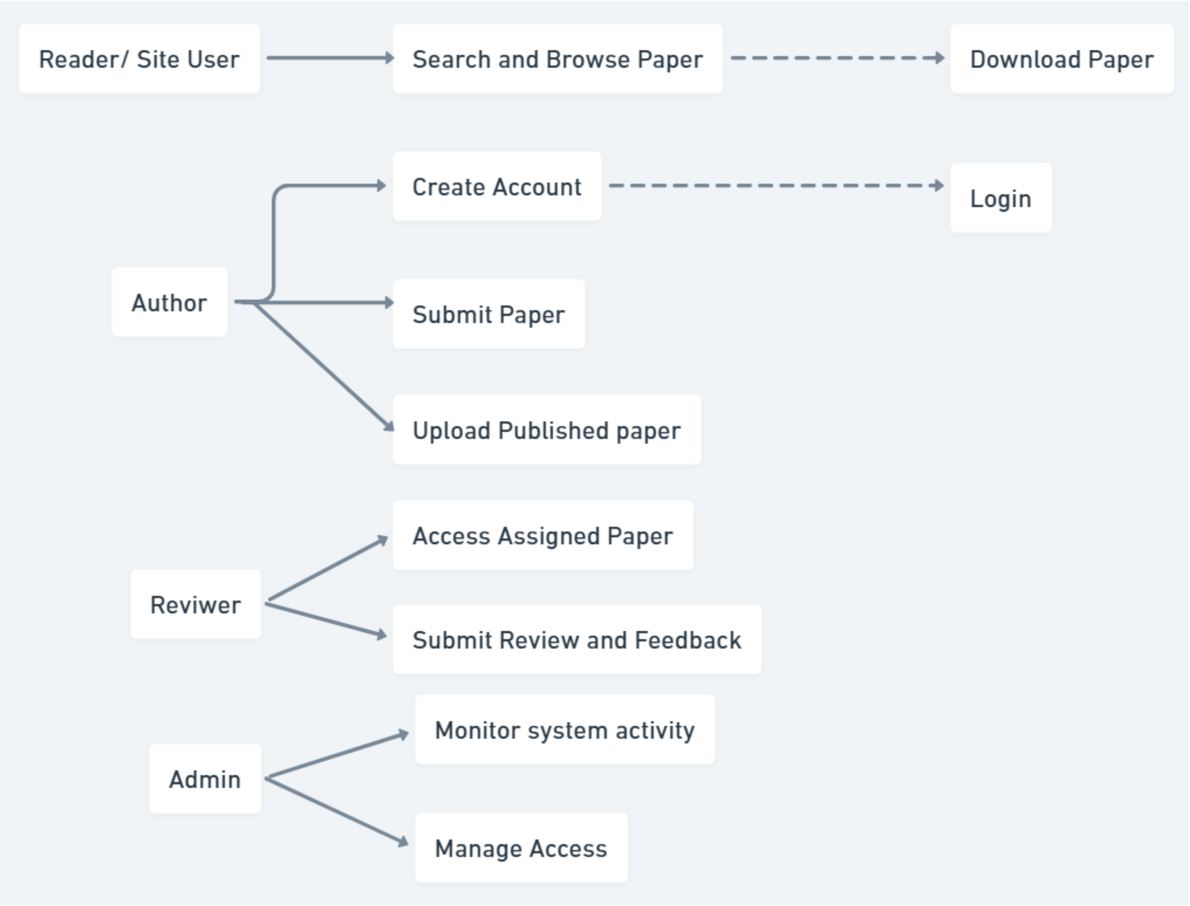
- Secure HTTPS protocol for all data transmissions.
- The system shall use SSL/TLS to encrypt all data transmitted over the network.
- The system shall support modern encryption protocols and be regularly updated to mitigate vulnerabilities.

3.4 Hardware Interfaces

No specific hardware interfaces are required beyond standard web hosting and server infrastructure.

4. Analysis Models

Use Case Diagram :



5. System Features

5.1 Author Registration and Profile Management

5.1.1 Description

- This feature allows authors to create and manage their profiles, update personal and academic information, and track their submissions.

5.1.2 Stimulus/Response Sequences

1. **Stimulus**: Author accesses the registration page.
   - **Response**: System displays a registration form with fields for name, affiliation, email, password, and research interests.
2. **Stimulus** : Author submits the completed form.
   - **Response** : System validates the input data, sends a verification email, and displays a confirmation message.
3. **Stimulus** : Author updates their profile information.
   - **Response** : System saves changes and updates the author's profile page.

5.1.3 Functional Requirements

- **REQ-4.1.1**: The system shall allow new authors to register by providing their name, affiliation, contact information, and research interests.
- **REQ-4.1.2**: The system shall verify the author's email address through a confirmation link.
- **REQ-4.1.3**: The system shall enable authors to update their biography, research interests, and publication history.
- **REQ-4.1.4**: The system shall display an error message for invalid inputs (e.g., incorrect email format).
- **REQ-4.1.5**: The system shall allow authors to delete their profiles.

5.2 Paper Submission and Management

5.2.1 Description

- This feature allows authors to submit papers for conferences or journals and manage the details of their submissions.

5.2.2 Stimulus/Response Sequences

1. **Stimulus**: Author selects the "Submit Paper" option.
   - **Response**: System displays a submission form with fields for paper title, abstract, keywords, category, and co-authors.
2. **Stimulus**: Author uploads a paper file in PDF or DOCX format.
   - **Response**: System checks the file format and size, then uploads the paper and displays a success message.
3. **Stimulus**: Author views the status of their submitted paper.
   - **Response**: System shows the status (submitted, under review, published) and any reviewer comments.

5.2.3 Functional Requirements

- **REQ-4.2.1**: The system shall allow authors to submit papers by filling out a submission form.
- **REQ-4.2.2**: The system shall support multiple file formats for uploads, including PDF and DOCX.
- **REQ-4.2.3**: The system shall provide a status tracker for submitted papers, including "submitted," "under review," and "published."
- **REQ-4.2.4**: The system shall allow authors to view reviewer feedback.

5.3 Reviewer Management

5.3.1 Description

- This feature allows reviewers to manage their profiles, receive assigned papers, and submit reviews.

5.3.2 Stimulus/Response Sequences

1. **Stimulus**: Reviewer logs into their account.
    - **Response**: System displays the reviewer's dashboard with assigned papers.
2. **Stimulus**: Reviewer opens a paper assigned for review.
    - **Response**: System displays the paper along with a review form to provide feedback and ratings.
3. **Stimulus**: Reviewer submits the completed review.
    - **Response**: System saves the review and notifies the author.

5.3.3 Functional Requirements

- **REQ-4.3.1**: The system shall allow reviewers to register and create a profile with their expertise.
- **REQ-4.3.2**: The system shall assign papers to reviewers based on their expertise.
- **REQ-4.3.3**: The system shall allow reviewers to view papers assigned to them.
- **REQ-4.3.4**: The system shall provide a form for reviewers to submit feedback and recommendations.

5.4 Search and Retrieval

5.4.1 Description

- Provides tools for readers to search and filter research papers based on various criteria.

5.4.2 Stimulus/Response Sequences

1. **Stimulus**: Reader enters a query in the search bar.
    - **Response**: System displays a list of relevant papers matching the query.
2. **Stimulus**: Reader applies filters for category and author.
    - **Response**: System updates the search results based on the applied filters.
3. **Stimulus**: Reader selects a paper from the search results.
    - **Response**: System displays the paper's details and download options.

5.4.3 Functional Requirements

- **REQ-4.4.1**: The system shall provide a search bar for querying papers by author, category, and keywords.

- **REQ-4.4.2**: The system shall support advanced search options with multiple filters
- **REQ-4.4.3**: The system shall display search results in a list format.
- **REQ-4.4.4**: The system shall allow readers to view detailed information about a selected paper.
- **REQ-4.4.5**: The system shall provide options to download papers in available formats.

5.5 Journals and Conferences Management

5.5.1 Description

- Manages dedicated pages for each journal or conference.

5.5.2 Stimulus/Response Sequences

1. **Stimulus**: User selects a specific journal or conference from the list.
   - **Response**: System displays a dedicated page with all associated papers.
2. **Stimulus**: Author submits a paper to a specific journal or conference.
   - **Response**: System processes the submission and updates the author's dashboard.

5.5.3 Functional Requirements

- **REQ-4.5.1**: The system shall provide dedicated pages for each journal and conference.
- **REQ-4.5.2**: The system shall list all papers associated with a journal or conference on its dedicated page.
- **REQ-4.5.3**: The system shall support direct submissions to journals and conferences.
- **REQ-4.5.4**: The system shall display journal/conference-specific metrics such as impact factor and acceptance rate.

6. Non Functional Requirements

6.1 Performance Requirements

- Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query
- The system shall handle up to 1,000 concurrent users.
- The system shall be dynamic in nature and update data.
- All papers shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.

6.2 Safety Requirements

No safety requirements have been identified.

6.3 Security Requirements

- All user data shall be encrypted during transmission.
- Implement multi-factor authentication for authors and reviewers.
- Regular backups of the database and system files.
- Ensure the system complies with GDPR and other relevant data privacy laws.

6.4 Software Quality Attributes

- **Adaptability**: Customizable for different conferences and journals
- **Availability**: 99% uptime annually
- **Correctness**: 100% compliance with functional specs; validated through unit and integration tests.
- **Portability**: Deployable on any cloud or on-premise
- **Reliability**: Supports up to 1000 users
- **Usability**: Intuitive and User friendly interface

Appendix A: Glossary

Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **ISSN**: International Standard Serial Number
- **UI**: User Interface
- **API**: Application Programming Interface
- **GPDR:** General Data Protection Regulation

Appendix B: Requirement Traceability Matrix

Requirement ID	Requirement Description	Design Specification	Implementation	Test Cases
REQ-4.1.1	The system shall allow new authors to register by providing their name, affiliation, contact information, and research interests.	Registration Form (UI-1), User Database (DB-1)	User Authentication Module (AUTH-1)	TC-1: Verify author registration process
REQ-4.1.2	The system shall verify the author's email address through a confirmation link.	Email Service Integration (EM-1)	Email Verification Module (EMAIL-VER-1)	TC-2: Verify email confirmation link
REQ-4.1.3	The system shall enable authors to update their biography, research interests, and publication history.	Profile Management Interface (UI-2)	Profile Update Function (PROFILE-UPDATE-1)	TC-3: Verify profile update functionality
REQ-4.1.4	The system shall display an error message for invalid inputs (e.g., incorrect email format).	Validation Script (VALID-1)	Input Validation Module (INPUT-VAL-1)	TC-4: Verify error messages for inputs
REQ-4.1.5	The system shall allow authors to delete their profiles.	Profile Management Interface (UI-2)	Profile Deletion Module (PROFILE-DEL-1)	TC-5: Verify profile deletion
REQ-4.2.1	The system shall allow authors to submit papers by filling out a submission form.	Submission Form (UI-3), User Database (DB-1)	Paper Submission Module (SUBMIT-1)	TC-6: Verify paper submission process
REQ-4.2.2	The system shall support multiple file formats for uploads, including PDF and DOCX.	File Upload Service (FILE-UPL-1)	File Handling Module (FILE-HANDLER-1)	TC-7: Verify file format compatibility
REQ-4.2.3	The system shall provide a status tracker for submitted papers, including "submitted," "under review," and "published."	Status Tracker Interface (UI-4)	Paper Tracking Module (TRACK-1)	TC-8: Verify status tracking accuracy
REQ-4.2.4	The system shall allow	Feedback Display	Feedback Module	TC-9: Verify

	authors to view reviewer feedback.	Interface (UI-5)	(FEEDBACK-1)	feedback visibility
REQ-4.3.1	The system shall allow reviewers to register and create a profile with their expertise.	Registration Form (UI-1), User Database (DB-2)	Reviewer Authentication Module (REVIEWER-AUTH-1)	TC-10: Verify reviewer registration
REQ-4.3.2	The system shall assign papers to reviewers based on their expertise.	Reviewer Assignment Algorithm (ALGO-1)	Paper Assignment Module (ASSIGN-1)	TC-11: Verify paper assignment process
REQ-4.3.3	The system shall allow reviewers to view papers assigned to them.	Reviewer Dashboard (UI-6)	Paper Display Module (DISPLAY-1)	TC-12: Verify paper visibility for reviewer
REQ-4.3.4	The system shall provide a form for reviewers to submit feedback and recommendations.	Feedback Form (UI-7)	Feedback Submission Module (FEEDBACK-SUB-1)	TC-13: Verify feedback submission process
REQ-4.4.1	The system shall provide a search bar for querying papers by author, category, and keywords.	Search Interface (UI-8)	Search Module (SEARCH-1)	TC-14: Verify search functionality
REQ-4.4.2	The system shall support advanced search options with multiple filters.	Advanced Search Interface (UI-9)	Advanced Search Module (ADV-SEARCH-1)	TC-15: Verify advanced search options
REQ-4.4.3	The system shall display search results in a list format.	Results Display Interface (UI-10)	Result Display Module (RESULT-DISP-1)	TC-16: Verify search result format
REQ-4.4.4	The system shall allow readers to view detailed information about a selected paper.	Paper Detail Interface (UI-11)	Paper Detail Module (PAPER-DETAIL-1)	TC-17: Verify paper detail visibility
REQ-4.4.5	The system shall provide options to download papers in available formats.	Download Service (DOWNLOAD-1)	Download Module (DOWNLOAD-MOD-1)	TC-18: Verify download functionality
REQ-4.5.1	The system shall provide dedicated pages for each journal and conference.	Journal/Conference Page Template (UI-12)	Journal/Conference Module (JOURNAL-CONF-1)	TC-19: Verify dedicated pages for journals/conferences
REQ-4.5.2	The system shall list all papers associated with a journal or conference on its dedicated page.	Journal/Conference Page Template (UI-12)	Paper Listing Module (LIST-1)	TC-20: Verify paper listing

REQ-4.5.3	The system shall support direct submissions to journals and conferences.	Submission Form (UI-3)	Direct Submission Module (DIRECT-SUB-1)	TC-21: Verify direct submission
REQ-4.5.4	The system shall display journal/conference-specific metrics such as impact factor and acceptance rate.	Metrics Display Interface (UI-13)	Metrics Module (METRICS-1)	TC-22: Verify metric display
NFR-5.1.1	Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query.	Performance Optimization (PERF-1)	Performance Module (PERF-MOD-1)	TC-23: Verify response time
NFR-5.1.2	The system shall handle up to 1,000 concurrent users.	Scalability Design (SCAL-1)	Load Balancing Module (LOAD-BAL-1)	TC-24: Verify concurrency handling
NFR-5.1.3	All papers shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.	Network Optimization (NET-OPT-1)	Download Performance Module (DL-PERF-1)	TC-25: Verify download speed
NFR-5.3.1	All user data shall be encrypted during transmission.	Security Protocol (SEC-1)	Data Encryption Module (ENC-1)	TC-26: Verify data encryption
NFR-5.3.2	Implement multi-factor authentication for authors and reviewers.	Authentication Design (AUTH-DES-1)	MFA Module (MFA-1)	TC-27: Verify multi-factor authentication
NFR-5.3.3	Regular backups of the database and system files.	Backup Strategy (BACKUP-1)	Backup Module (BACKUP-MOD-1)	TC-28: Verify backup process
NFR-5.3.4	Ensure the system complies with GDPR and other relevant data privacy laws.	Compliance Checklist (COMP-1)	Compliance Module (COMP-MOD-1)	TC-29: Verify GDPR compliance

PES UNIVERSITY

Department of Computer Science

and Engineering

UE22CS341A: Software

Engineering

Software Architecture Document for Conference Paper Management System

Name : Vedant Varma
SRN : PES1UG22CS681

Name : Vivek M
SRN : PES1UG22CS707

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

Purpose

This document outlines the requirements for the Conference Paper Management System. The system will serve as an intermediary application, facilitating interaction between authors, reviewers, and readers. The system allows authors to submit papers, reviewers to manage reviews, and readers to browse and search for published papers. This document is intended for both the stakeholders and the developers of the system.

Scope

The system is a comprehensive platform for managing academic publications, designed to streamline interactions among authors, reviewers, and readers. Authors can upload their manuscripts, provide detailed metadata, and manage revisions through the system. It supports the entire submission process, including tracking paper updates and handling supplementary materials. Reviewers are integral to the peer review process, using the system to provide structured feedback, assess paper quality, and communicate with authors. Readers benefit from advanced search and browsing features, allowing them to discover and access published content easily. The system maintains extensive databases for papers, authors, and review data, ensuring efficient retrieval and secure management of sensitive information.

Definitions, Acronyms and Abbreviations

- **SRS**: Software Requirements Specification
- **ISSN**: International Standard Serial Number
- **UI**: User Interface
- **API**: Application Programming Interface
- **GPDR:** General Data Protection Regulation

References

- IEEE Xplore (https://ieeexplore.ieee.org/Xplore/home.jsp)
- EasyChair (https://easychair.org/)
- IEEE Standard for Software Requirements Specifications (IEEE Std 830 1998)
- The “4+1” view model of software architecture, Philippe Kruchten, November 1995 (http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf)

2. Architectural Representation

This document details the architecture using the views defined in the “4+1” model [Krutchen’s View Point]. The **4+1 model** organizes the architecture into five distinct views, each addressing specific aspects of the system's design and implementation.

Logical View

Audience: Designers
Area: Functional Requirements: Focuses on the design's object model. Captures the most important use-case realizations and object interactions.
Related Artifacts: Design Model (e.g., class diagrams, sequence diagrams).
Key Considerations: Object relationships, their interactions, and key use cases.

Process View

Audience: Integrators
Area: Non-functional Requirements: Focuses on concurrency, synchronization, and performance aspects.
Related Artifacts: Design patterns for concurrency (such as message queues, thread pools) may be reflected in the design model.

Implementation View

Audience: Programmers
Area: Describes the layers, modules, and subsystems of the system.
Related Artifacts: Implementation Model (e.g., component diagrams, module breakdown).
Key Considerations:

- How components are developed, organized, and interact with each other.
- Layering strategies (e.g., presentation, business logic, data access layers).

Deployment View

Audience: Deployment Managers
Area: Topology of the system: Focuses on the physical layout of the system, including hardware and how software maps onto this hardware.
Related Artifacts: Deployment Model (e.g., network diagrams, server setups).

Use Case View

Audience: All stakeholders, including end-users
Area: Core Functionality: Describes the main functionality of the system through scenarios or use cases.
Related Artifacts: Use-Case Model (use case diagrams, narratives).
Key Considerations: Critical system functionality from the user’s perspective.

Data View

Audience: Data Specialists, Database Administrators
Area: Persistence and Data Management: Describes the persistence aspects of the system, including databases and storage mechanisms.
Related Artifacts: Data Model (e.g., entity-relationship diagrams, schema designs).
Key Considerations: Data storage, retrieval mechanisms, database design.

3. Architectural Goals and Constraints

Goals

- Provide a platform for **efficient paper management** that streamlines the submission, review, and publishing process.
- Ensure **scalability** to handle multiple conferences and journals with high volumes of papers and users.
- Deliver a **secure** environment for user data, ensuring GDPR compliance and secure data transmission.

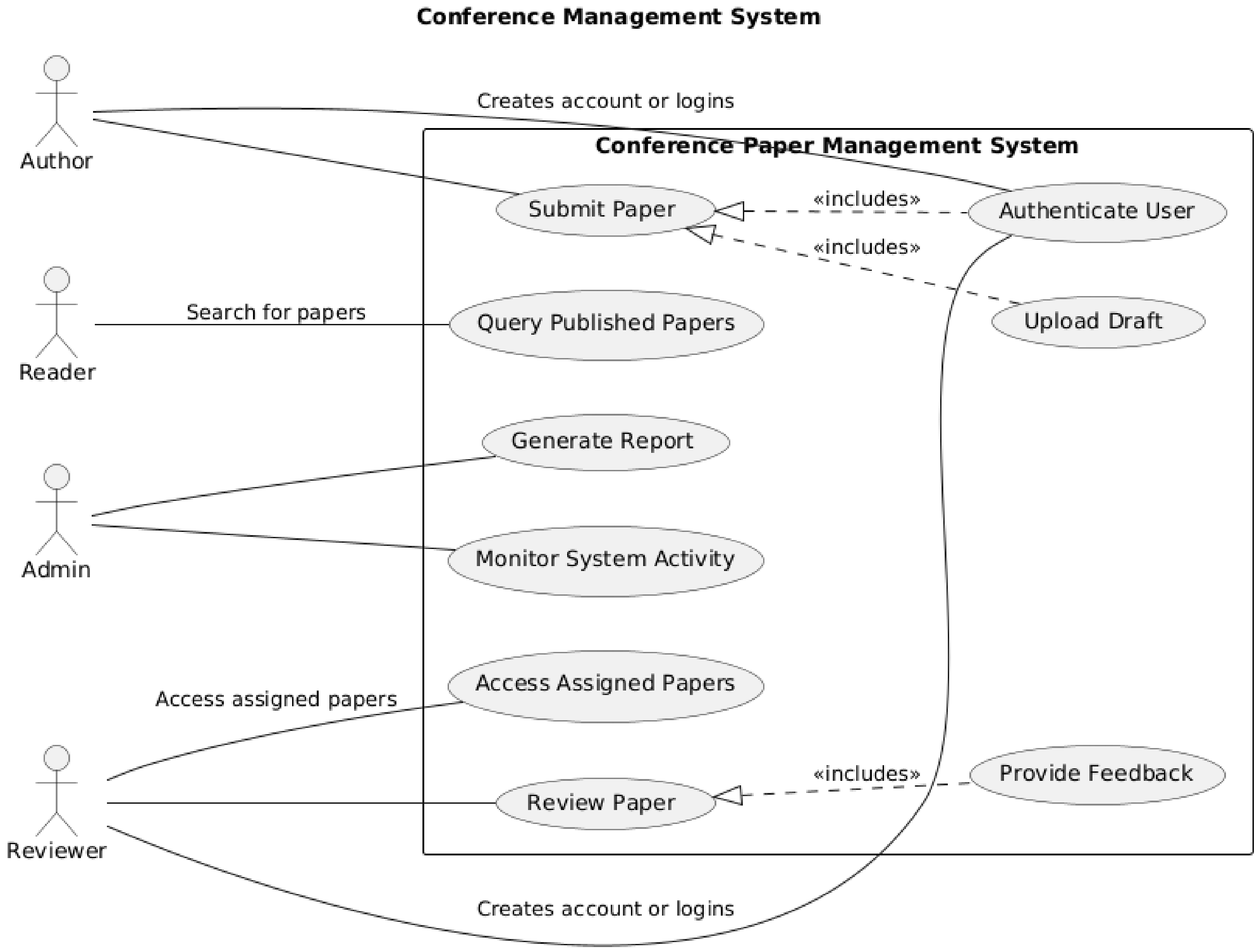
Constraints

- **Security**: All data transmission must be encrypted, and the system must implement multi-factor authentication for users (authors and reviewers). GDPR compliance is mandatory.
- **Scalability**: The system should scale to handle **up to 1,000 concurrent users** and multiple conferences simultaneously.
- **Performance**: Response times for searches and submissions must be within 7 seconds for optimal user experience, and download times should not exceed 10 seconds over low-bandwidth connections.

4. Use Case View

The primary use-cases include :

- **Author Registration and Paper Submission:** Authors register on the platform, update profiles, submit papers to journals or conferences, and track paper status.
- **Reviewer Assignment and Paper Review:** Reviewers log in to view assigned papers, submit feedback, and track their review history.
- **Reader Search and Retrieval:** Readers use the search tool to discover and download papers based on author, category, and keywords.
- **Journal and Conference Management:** Authors submit papers directly to specific conferences or journals, and the system tracks each conference's papers and statistics.



5. Logical View

The logical architecture highlights the core components and their interactions:

Modules

- User Management: Handles author/reviewer/reader registration, login, profile updates, and account deletion.
- Paper Submission and Tracking: Manages the submission process, including file uploads, metadata input, and status updates.
- Review System: Enables reviewers to view assigned papers, provide feedback, and track review history.
- Search and Retrieval: Allows users to query the paper database and retrieve relevant results.
- Journal/Conference Management: Manages specific journal/conference pages, paper listings, and submission tracking.

Data Flow

- Users submit data (e.g., paper submissions, reviews) through the UI.
- Data is processed by the backend (Flask), which communicates with the MySQL database for storage and retrieval.
- Responses are sent back to the frontend for display (e.g., submission confirmation, search results).

Layering

Presentation Layer: The user interface components that interact with the users (authors, reviewers, and readers). This could be web or mobile apps.

- **UI/Frontend**: This is where users submit papers, track submissions, view assigned reviews, and search for papers. It includes web forms, buttons, and search bars.

Application/Service Layer: The business logic layer, where the core functionality is implemented.

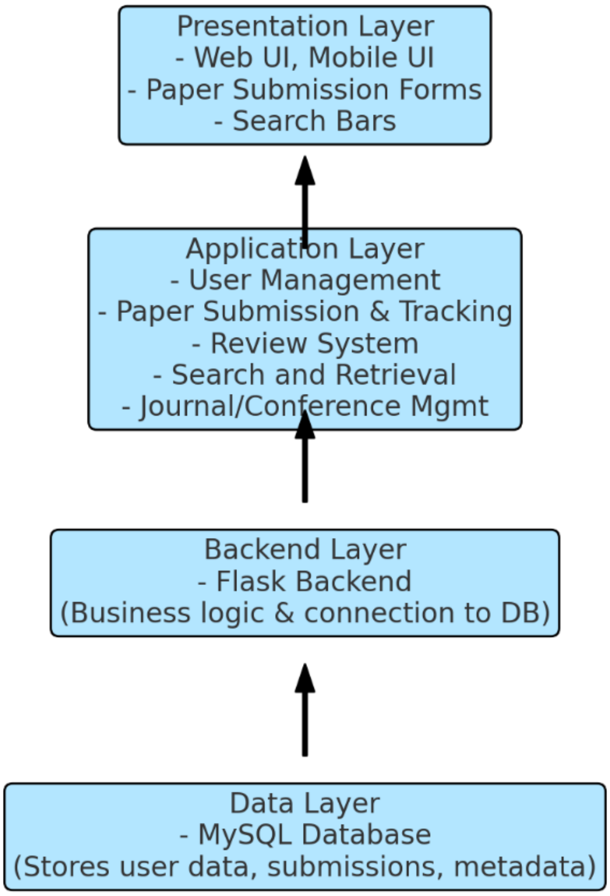
- **User Management Module**: Handles user-related actions such as registration, login, profile updates, and deletion.
- **Paper Submission and Tracking Module**: Manages file uploads, metadata input, and submission status.
- **Review System Module**: Allows reviewers to view papers and provide feedback.
- **Search and Retrieval Module**: Queries the paper database and retrieves results.
- **Journal/Conference Management Module**: Manages conference pages, paper listings, and submission status.

Database Layer: The data storage and retrieval layer where the persistent data lives.

- **MySQL Database**: Stores user data, paper submissions, reviews, metadata, and conference/journal details.

Backend/Controller Layer: Bridges the application layer and the database layer.

- **Flask Backend**: Processes user requests, handles business logic, and communicates with the MySQL database



6. Process View

This section describes the processes, their allocation to design elements, and how they relate to the overall system model and implementation.

Processes

The Conference Paper Management System consists of multiple processes:

- **Author Registration and Profile Management**: Handles the creation, updating, and management of author profiles.
- **Paper Submission and Management**: Allows authors to submit papers, track their submission status, and make updates to their manuscripts.
- **Reviewer Assignment and Management**: Responsible for assigning papers to reviewers based on their expertise.
- **Search and Retrieval**: Enables users (readers) to search for and retrieve academic papers based on different criteria, such as keywords, author names, and paper categories.
- **Journals and Conferences Management**: Manages the association between papers and specific journals or conferences.

Process to Design Elements

Each process in the system is associated with a set of design elements. These design elements encapsulate the functional logic and user interactions.

- Author/ reviewer registration is mapped to a profile_management_interface associated with a user database
- All users and readers are handled by a robust user interface having search modules to effectively search and retrieve papers
- Submission_Forms and File_Upload_services are used to handle paper submission and management.
- Reviewer Assignment and Management is Mapped to a reviewer dashboard, with a feedback_submission module
- **Journals and Conferences Management**: Mapped to the JournalConferencePage for displaying journals and conferences, and the PaperListingModule for listing associated papers and metrics.

Model Dependencies

The system’s processes depend on various external and internal components for smooth execution. The major dependencies are:

- **Database and Query Dependency**: Most processes, such as user registration, paper submission, and reviewer management, rely heavily on the database for persistent storage and retrieval of data.
- **Reviewer Assignment Algorithm**: The **Reviewer Assignment Process** is dependent on a paper assignment algorithm that assigns each uploaded manuscript to appropriate reviewers accordingly

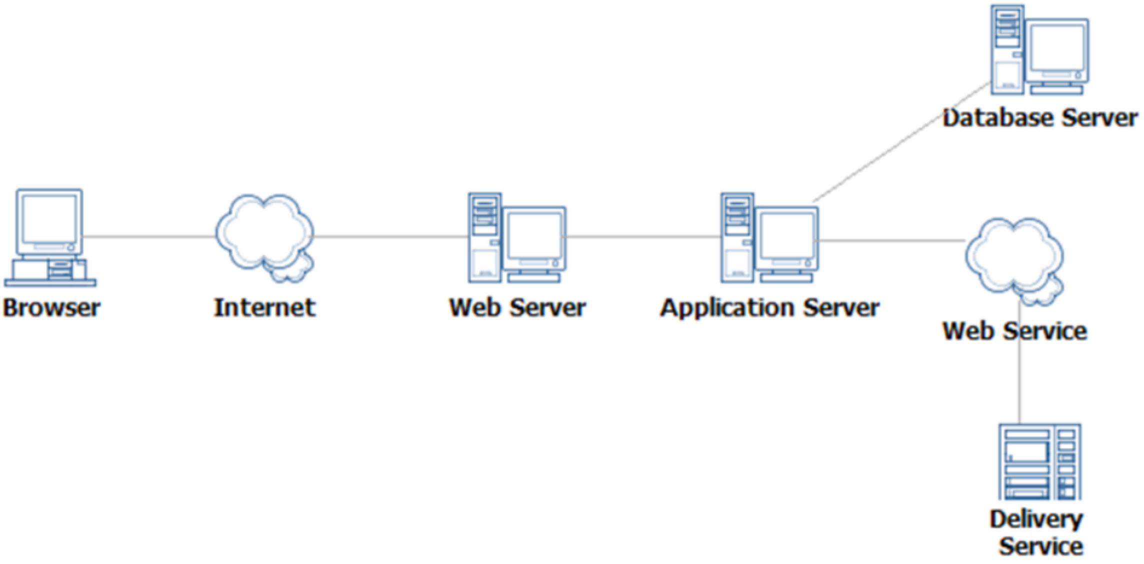
Processes to the Implementation (Explanation of DFD)

The **Conference Paper Management System (CPMS)** DFD outlines interactions between users (Authors, Reviewers, Readers, Admins) and system processes.

- **Authors** authenticate, upload paper drafts, and track submissions. Paper submissions are stored in the **Papers Database** and are assigned to **Reviewers** based on expertise or conference affiliation.
- **Reviewers** log in to access assigned papers, provide feedback, and approve or reject submissions. The feedback is stored in the system.
- Once a paper is accepted, it moves to **Proceedings**, making it available for **Readers** to query and access based on specific filters.
- **Admins** monitor the system via **Report Logs** that track submission status, review activity, and overall system performance.

Data is stored across various components, including **Author Info**, **Reviewer Info**, **Conference Info**, and **Proceedings Info**. **Readers** can search and retrieve published papers, while **Admins** oversee the submission and review process through generated reports. The DFD highlights the flow of data from paper submission to publication, connecting key entities and ensuring a streamlined review and publication process

7. Deployment View



- ***External Desktop PC***:
  - ***Users***: *Authors, reviewers, and readers.*
  - ***Interaction***: *Communicates with the web server via HTTPS for paper submission, review management, and accessing published papers.*
- ***Desktop PC (Admin)***:
  - ***Users***: *Admin and support staff.*
  - ***Interaction***: *Access through a dedicated internal interface for managing user accounts, paper submissions, system analytics, and maintenance.*
- ***Web Server***:
  - ***Functionality***: *Hosts the web interface for users.*
- ***Application Server (Flask)***:
  - ***Functionality***: *Handles business logic, processes requests, and communicates with the database server.*
  - ***Technology***: *Flask (Python-based), RESTful APIs.*
- ***Database Server (MySQL)***:
  - ***Functionality***: *Stores user profiles, paper submissions, reviews, and system logs.*
  - ***Technology***: *MySQL with indexing for efficient data operations.*
  - ***Backup & Security***: *Regular backups, encrypted communication, and compliance with GDPR.*

8. Implementation View

Database:

- **MySQL** is used as the primary database.
- **Nested queries** and **correlated queries** will be employed for complex data retrieval.
- **Functions, procedures, and triggers** will be used to automate tasks and manage workflow efficiently.
- **Virtual tables** (views) will be created for simplified data access and aggregation.
- **Indexing** will be applied to frequently queried columns to optimize performance.

Backend:

- **Flask** is used as the backend framework.
  - It handles API requests, manages business logic, and interacts with the MySQL database.
  - **RESTful APIs** over HTTP/HTTPS is used for communicating with the front end framework
  - **Stored procedures** will be invoked from the backend to perform complex database operations, and **triggers** will automate backend processes like updating statuses or sending notifications.

Frontend:

- **HTML, CSS, JS** (or alternatively, **React**) is used for the frontend.
  - It provides a dynamic, responsive user interface that interacts with the backend.
  - Fetches and displays data using **API responses**, rendering user profiles, submission statuses, and search results in real-time.
  - Supports user interactions such as form submissions (e.g., login, paper submissions) and search queries that communicate with the backend.

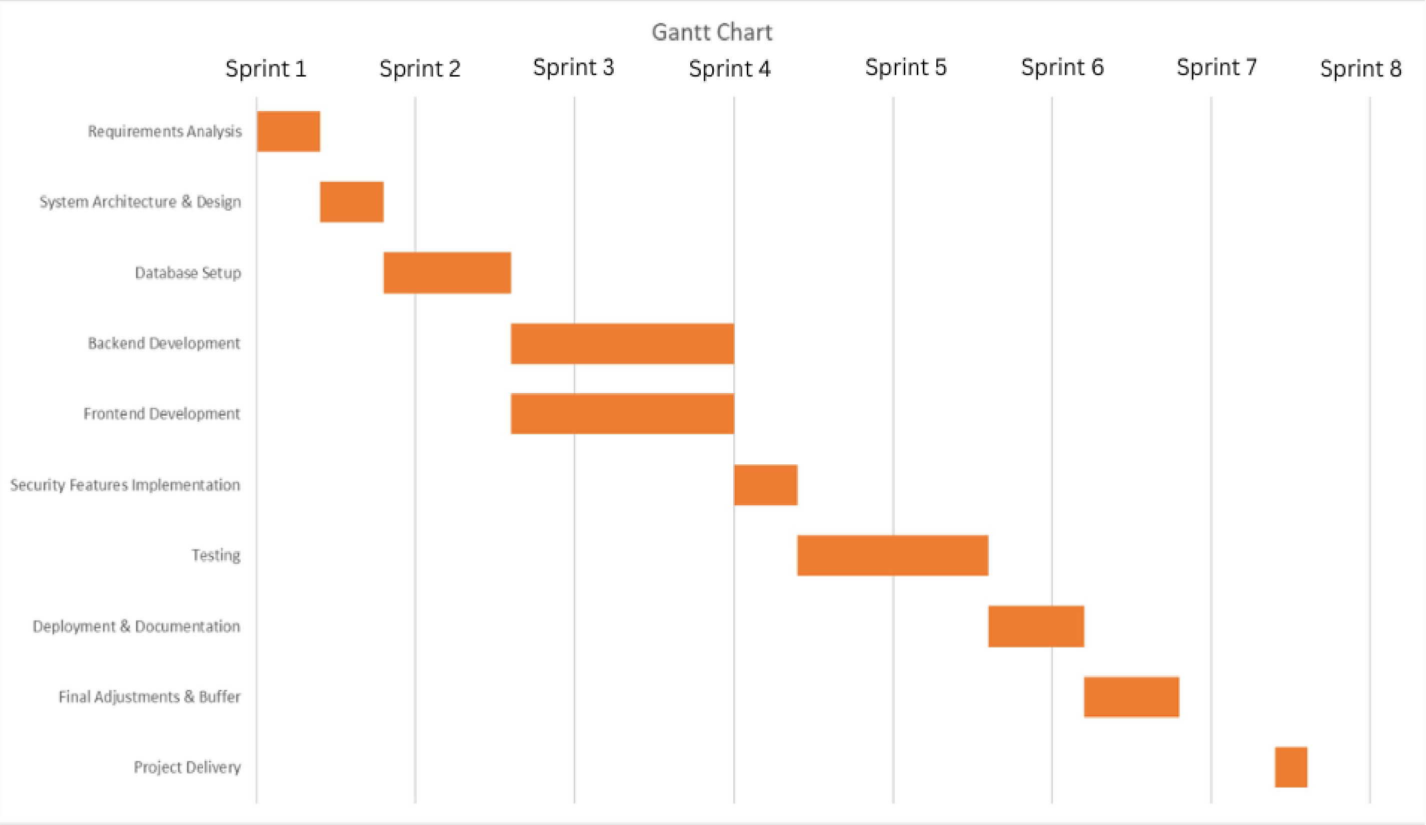
9. Performance

- Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query
- The system shall handle up to 1,000 concurrent users.
- The system shall be dynamic in nature and update data.
- All papers shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.

10. Quality

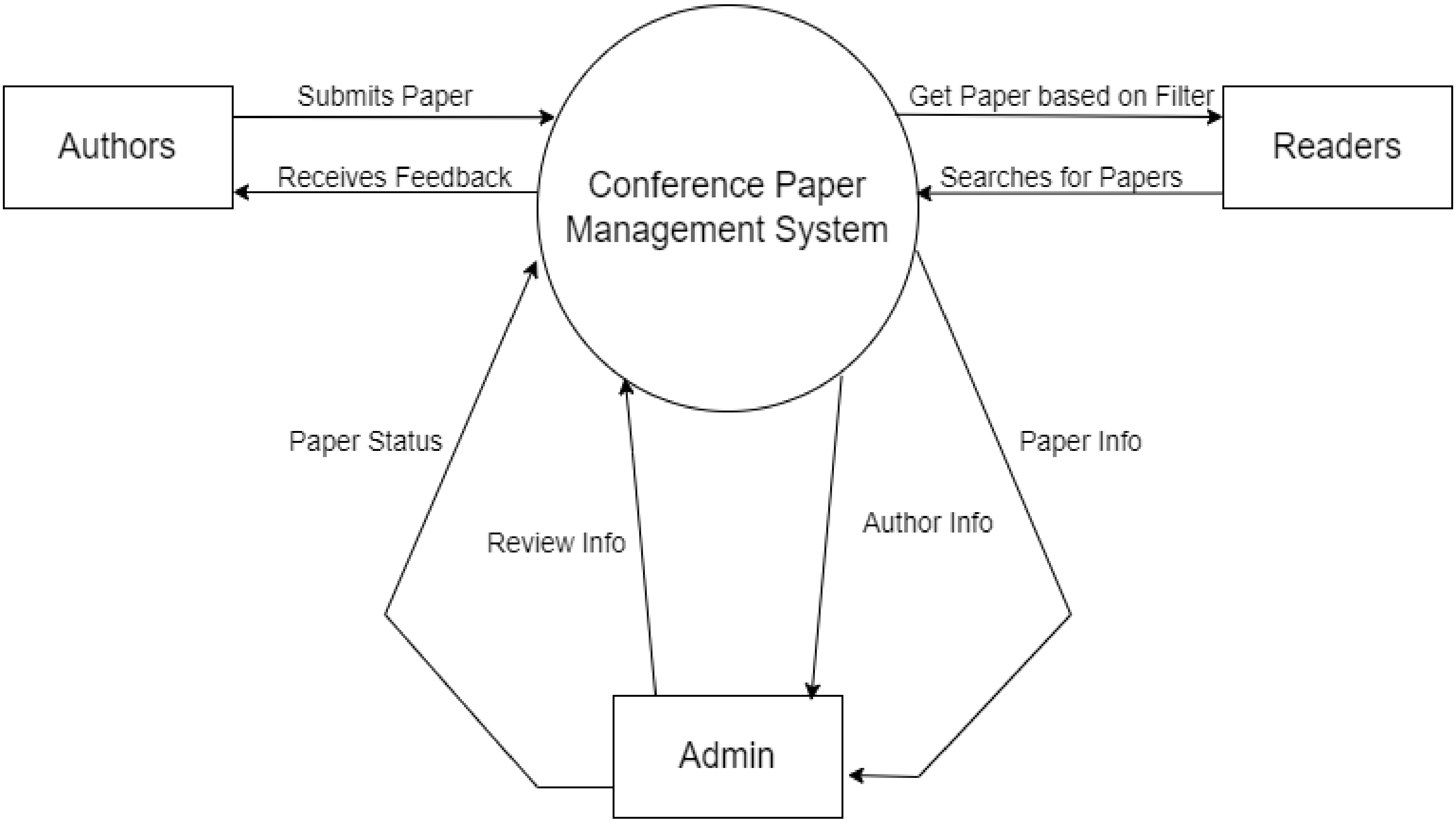
- **Adaptability**: Customizable for different conferences and journals
- **Availability**: 99% uptime annually
- **Correctness**: 100% compliance with functional specs; validated through unit and integration tests.
- **Portability**: Deployable on any cloud or on-premise
- **Reliability**: Supports up to 1000 users
- **Usability**: Intuitive and User friendly interface

11. Gantt Chart

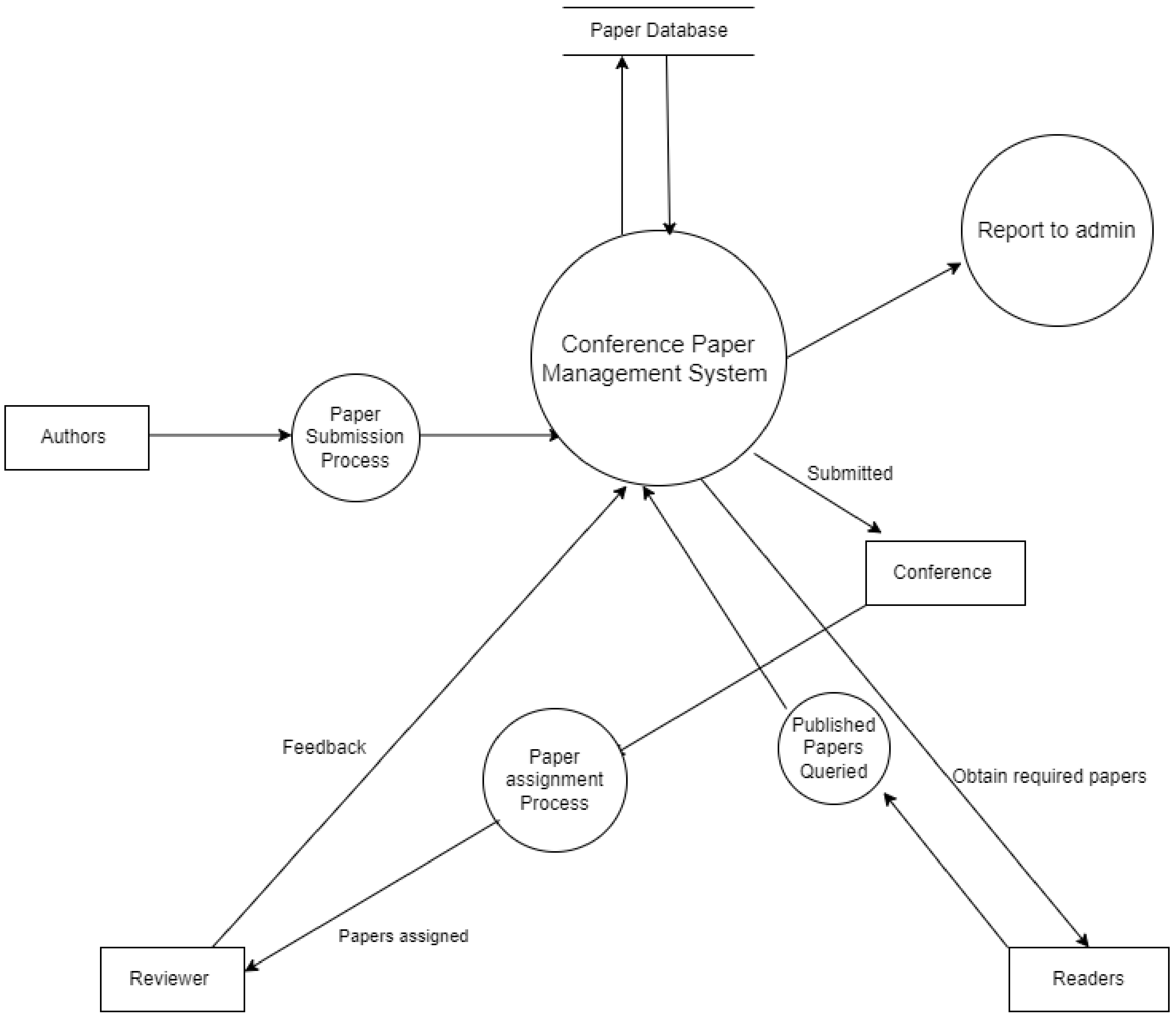


The process started with the creation of SRS which marks the Sprint 1 in 1st week of September.
There is a gap between Sprint 1 and Sprint 2 and the start of System Architecture and Design starts from 4th October.
Sprint 2 starts from October 13th and the following sprints are the subsequent weeks.

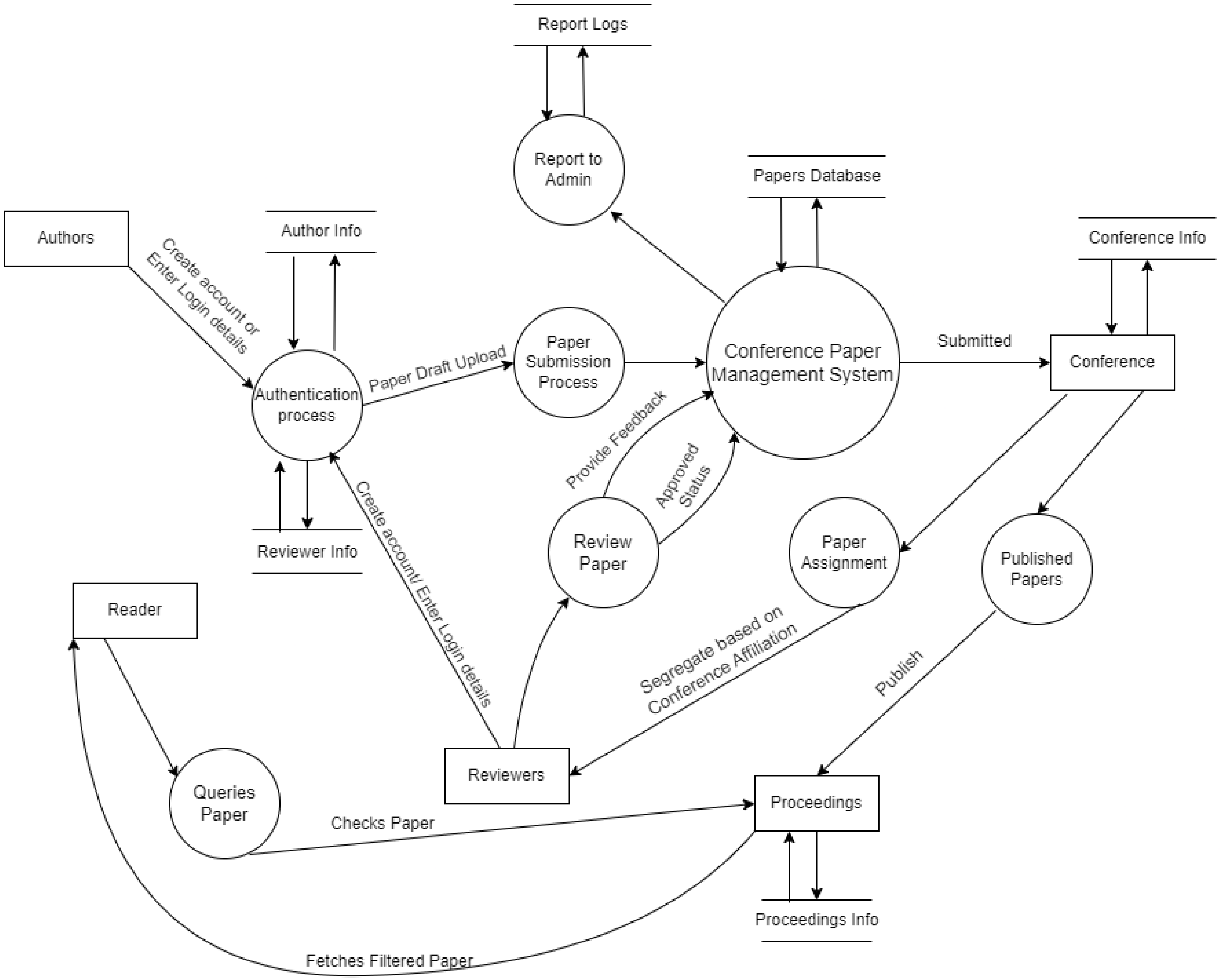
12. Data Flow Diagram Level 0



13. Data Flow Diagram Level 1



14. Data Flow Diagram Level 2



UE22CS341A: Software Engineering
Case Study

Software Testing Document for Conference Paper Management System

Name : Vedant Varma

SRN : PES1UG22CS681

Name : Vivek M

SRN : PES1UG22CS707

For SRS Document for Conference Paper Management System

Functional and Non-Functional Requirements:

System Features

- **Author Registration and Profile Management**:
  - Authors can create and manage profiles, update information, and track submissions.
- **Paper Submission and Management**:
  - Authors submit papers, manage details, and view submission statuses.
- **Reviewer Management**:
  - Reviewers manage profiles, receive papers, and submit reviews.
- **Search and Retrieval**:
  - Readers can search and filter research papers by various criteria.
- **Journals and Conferences Management**:
  - Dedicated pages for journals and conferences, allowing paper submissions and displaying metrics.

Non-Functional Requirements

- **Performance Requirements**:
  - Queries must load within 15 seconds; supports 10 concurrent users; papers should download within 10 seconds.
- **Safety Requirements**:
  - No specific safety requirements identified.
- **Security Requirements**:
  - User data encryption; GDPR compliance.
- **Software Quality Attributes**:
  - Adaptability, 99% availability, 100% correctness, portability, reliability (supports 1,000 users), and user-friendly interface.

1. Unit Testing

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Result	Test Result
UT-1	Author Registration	Validate author registration form fields	Registration page open	1. Access registration form 2. Fill in valid data and submit	Name: "John Doe", Email: "john@example.com"	Confirmation message and email verification sent.	Confirmation message sent	Pass
UT-2	Profile Management	Update author profile information	Author logged in	1. Access profile. 2. Update biography 3. Save changes	Biography: "Researcher in AI."	Changes saved, and updated profile displayed.	Changes saved, and updated profile displayed.	Pass
UT-3	Paper Submission	Validate paper submission file formats	Author logged in	1. Access submission form 2. Upload a PDF file and Submit	File: "paper.pdf"	Success message and file uploaded.	Success message and file uploaded.	Pass
UT-4	Search and Retrieval	Validate search query functionality	Search page open	1. Enter query in search bar. 2. Click search.	Query: "AI in healthcare"	List of relevant papers displayed.	List of relevant papers displayed.	Pass
UT-5	Reviewer Management	Validate reviewer profile creation	Reviewer registration page open	1. Fill in profile data. 2. Submit registration.	Name: "Jane Smith", Expertise: "ML"	Confirmation message and profile created.	Confirmation message and profile created.	Pass
UT-6	Journals and Conferences	Validate journal submission process	Conference Page Open	1. Select journal. 2. Check proceedings	Paper title: "New Trends in AI"	Submission processed and updated dashboard.	Submission processed and updated dashboard.	Pass

2. Integration Testing

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Result	Test Result
IT-1	Author Registration & Email	Verify email verification process	Author registered	1. Submit registration. 2. Click verification link in email.	Email: "john@example.com"	Author account activated and logged in.	Author account activated and logged in.	Pass
IT-2	Paper Submission & Review	Validate paper submission to review process	Author submitted paper	1. Reviewer logs in. 2. Opens assigned paper. 3. Submits review.	Reviewer comments: "Great paper!"	Review saved and author notified.	Review saved and author notified.	Pass
IT-3	Search & Filter	Test integration of search with filtering	Search page open	1. Enter search term. 2. Apply filters. 3. Execute search.	Author: "Jane Smith", Category: "AI"	Filtered results displayed correctly.	Filtered results displayed correctly.	Pass
IT-4	Profile Management & Updates	Verify profile update reflected in submissions	Author logged in	1. Update profile 2. Submit a paper 3. Check submission details.	Updated biography	Submission reflects updated author information.	Submission reflects updated author information.	Pass
IT-5	Conference Management & Papers	Validate paper listing under specific conference	Conference selected	1. Select conference. 2. View associated papers.	Conference: "AI Summit"	List of papers associated with the conference displayed.	List of papers associated with the conference displayed	Pass

3. System Testing

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Result	Test Result
ST-1	Overall System Functionality	Validate overall system performance	System deployed	1. Perform concurrent logins. 2. Submit papers simultaneously.	3 concurrent users	System handles multiples users without crashing.	System can't multiple users	Fail
ST-2	Security Features	Test data encryption during transmission	User data submitted	1. Submit data. 2. Inspect data transmission.	User data	User data encrypted during transmission	User data encrypted during transmission	Pass
ST-3	Usability Testing	Assess user interface intuitiveness	User engaged	1. Conduct user sessions. 2. Collect feedback on navigation.	User demographics	Users find the interface intuitive and easy to navigate.	Easy UI	Pass
ST-4	Performance Testing	Test response times for queries	System deployed	1. Submit various queries. 2. Measure response times.	Queries: Various	All queries respond within 15 seconds.	All queries respond within 10 seconds.	Pass