LOOMIO

"Threads of effort, woven into outcomes"

For the partial fulfillment of the requirements for the completion of the course - Software Engineering Practice (CS307)

SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)

Team Members:

Y Sri Rama Bharadwaj – 123cs0030

J V Kousthub – 123cs0074

S Sai Videsh – 523cs0014

P Sanjay Davis – 123cs0031

1. Description of the Proposed Software Product

Loomio is a **web-based Community-Based Task Management System** designed to facilitate structured collaboration, accountability, and productivity within organizations, student groups, volunteer communities, and professional circles. Unlike simple task trackers, Loomio integrates multiple dimensions of group participation—task delegation, contribution assessment, attendance management, and event scheduling—into a **single, unified platform**. The primary goal is to ensure that collaborative work not only gets done but is also measured, recognized, and transparently displayed.

At its core, Loomio addresses a recurring issue in group work: **ambiguity in responsibility and recognition.** In traditional systems, tasks are often delegated informally, making it difficult to determine individual contributions or track progress systematically. Loomio overcomes this by providing **role-based task assignment**, **real-time progress tracking**, and **quantifiable contribution points** that reward active participation. These features make Loomio suitable not only for academic communities but also for NGOs, startups, and small organizations that require accountability without investing in expensive enterprise tools.

The system also integrates **attendance and leave management**, which is particularly relevant in community groups where participation is voluntary but still critical. Leaders can monitor participation trends, approve or reject leave requests, and ensure that accountability is maintained even in loosely structured organizations. Attendance logs feed directly into contribution analytics, giving leaders and members a holistic picture of involvement.

Another defining feature of Loomio is the **community calendar.** Many collaborative tools fail because they separate task management from scheduling. Loomio brings them together, ensuring that deadlines, events, and meetings appear in a **centralized calendar view** accessible to all members. This reduces miscommunication and prevents missed deadlines.

The platform is built on a **modern technology stack** that ensures scalability and ease of use:

- **Frontend:** Developed in **Vite + React**, ensuring fast performance and a clean, responsive interface.
- **Backend:** Powered by **Express.js (Node.js)**, which provides RESTful APIs for seamless communication between client and server.

- **Database:** A **MySQL** relational database that stores structured records of users, tasks, contributions, attendance, and events.
- **Authentication:** Secured with **JWT (JSON Web Tokens)** to enforce role-based permissions.
- **Notifications:** Will via **Nodemailer**, with support for SMTP services like Gmail or Mailgun.

From a usability standpoint, Loomio is designed for **minimal learning curves.** The interface is simple, with dashboards tailored to the needs of different users. For instance, an administrator's dashboard emphasizes approval workflows and contribution analytics, while a member's dashboard highlights assigned tasks, attendance logs, and personal progress summaries.

The product vision for Loomio can be summarized as follows:

- **Accountability:** Establish measurable records of participation and responsibility.
- **Transparency:** Ensure that tasks and contributions are visible to the entire community.
- **Accessibility:** Provide a platform that works across devices and is intuitive even for first-time users.
- **Affordability:** Deploy on free-tier or low-cost hosting platforms (Vercel, Railway) to keep costs minimal for student or volunteer groups.

In essence, Loomio is not just a tool for completing tasks - it is a **framework for fostering collaboration, recognizing effort, and sustaining engagement in communities.**

2. Major Inputs to the System

The inputs to Loomio are shaped by the **role of the user**. The platform differentiates between **administrators/leaders** and **regular members**, ensuring that each role interacts with the system in ways relevant to their responsibilities.

From the Administrator's Perspective:

1. **User Details:** Admins input member information such as names, email addresses, roles, and permissions. This ensures that each user has appropriate access rights. For example, only leaders can approve leave requests or assign contribution points.

2. **Task Information:** Admins create tasks, define priorities, set deadlines, and assign them to one or more members. Inputs here include task titles, descriptions, deadlines, and associated members.
3. **Attendance Data:** Leaders can mark attendance, record absences, and attach contextual notes. These entries serve as official participation records.
4. **Event Scheduling:** Admins schedule meetings, workshops, and deadlines. Input fields include event titles, dates, locations (if physical), or links (if online).
5. **System Settings:** Leaders configure notification preferences, determine visibility levels for reports, and adjust community-wide settings.

From the Member/User's Perspective:

1. **Profile Updates:** Members input personal details such as updated contact information, profile pictures, or communication preferences.
2. **Task Updates:** Users change task status (e.g., *Not Started* → *In Progress* → *Completed (Mark for review for admin)*), add progress notes, and optionally put supporting links.
3. **Event Participation:** Members confirm their attendance for scheduled events, propose alternate times, or provide feedback on scheduling.
4. **Feedback and Communication:** Users can post comments, ask clarifying questions, and provide feedback directly within task or event modules. This ensures discussions remain tied to specific activities instead of being lost in separate communication tools.

The **combination of administrative and member inputs** ensures that Loomio captures the complete lifecycle of collaboration, from the moment a task is created, through its execution, to its final completion and evaluation. Every input is stored in the MySQL database, forming the basis for generating reports, analytics, and dashboards.

3. Major Outputs of the System

Just as inputs are role-specific, the outputs in Loomio are **customized for different users**. These outputs provide actionable insights, enhance accountability, and ensure transparency.

Outputs for Administrators:

1. **Task Overview Reports:** The Task Overview Reports for admins now include a "Submission Link" column where each task marked as "Completed" has a clickable link to the submitted files or details provided by the user for review.

   This link can lead to a Google Drive, Dropbox folder, or an internal file storage system where the user uploaded their task proof (document, image, etc.).

   The admin can quickly check these links while evaluating the task progress..

2. **Attendance Reports:** Attendance summaries can be generated daily, weekly, or monthly. For example, an administrator can see which members attended 90% of meetings and which fell below participation thresholds.
3. **Event Calendar:** Admins see a full calendar of events, deadlines, and meetings. The calendar supports filtering by event type or member involvement.
4. **Contribution Analytics:** A detailed breakdown shows how many tasks each member completed, how many points were earned, and how consistent participation has been. This enables data-driven recognition of contributions.
5. **Alerts and Notifications:** Automated reminders are sent to highlight overdue tasks, pending approvals, or absenteeism. Admins receive system-level notifications for immediate action.

Outputs for Members:

1. **Personal Task List:** Each user sees a customized task dashboard listing their assignments, deadlines, and statuses. Tasks are prioritized and color-coded for clarity.
2. **Personal Attendance Summary:** Members can view their own attendance records, including approved leaves and days present/absent. This fosters self-awareness and accountability.
3. **Upcoming Events:** A personalized calendar highlights events relevant to the user. For example, if a meeting concerns only a subgroup, only those members see it in their calendar.
4. **Contribution Summary:** Members receive feedback in the form of contribution points and charts showing their progress over time. This encourages continuous improvement.
5. **Notifications:** Real-time notifications inform members about new assignments, changes to task deadlines, upcoming events, or leave approvals/rejections.

Outputs in Loomio are **not static reports** but **interactive dashboards.** Users can filter, drill down, and even export certain outputs into CSV or PDF for record-keeping. This dual functionality (interactive + exportable) makes Loomio equally valuable for day-to-day collaboration and long-term documentation.

4. Major Processing Functionality

Loomio's processing functionality is the **engine of the system**, responsible for transforming inputs into meaningful outputs. The software is designed to support **end-to-end workflows** around task management, attendance tracking, contribution assessment, and event scheduling. Each functionality area incorporates validation, secure data handling, and automation to reduce administrative burden.

Task Lifecycle Management

When an administrator creates a task, Loomio processes the request by:

1. **Validating input data** (title, description, deadline, assigned users).
2. **Recording task details in MySQL**, linked to the relevant users.
3. **Triggering notifications** to assigned members via email and system alerts.
4. **Tracking changes** in task status (Not Started → In Progress → Completed).
5. **Logging completion** and updating contribution points accordingly.

This flow ensures that every task has a clear owner, a visible status, and a measurable outcome.

Contribution Points Processing

Contribution assessment is a unique feature of Loomio. The system processes contributions by:

- **Allocating points** each time a member completes a task, attends an event, or contributes to discussions.
- **Applying weighted scoring** (e.g., major tasks = 10 points, attendance = 5 points, participation in discussions = 2 points).
- **Aggregating points** per user and updating analytics dashboards.
- **Normalizing scores** across members to ensure fairness.

This automated calculation ensures that recognition is consistent, objective, and free from manual bias.

Attendance and Leave Workflow

The system processes attendance in two key ways:

1. **Marking Presence/Absence** – Either by leader input or member self-check-in (if enabled).
2. **Leave Requests** – Processed by:
    - Logging the request in the database.
    - Notifying the relevant leader for approval.
    - Updating attendance records based on approval/rejection.

All attendance data is stored with timestamps, ensuring tamper-proof participation records.

Calendar & Event Scheduling

Event-related processing involves:

- **Storing event details** (title, description, date/time, type, participants).
- **Cross-linking events to tasks** when applicable (e.g., a task deadline tied to a specific meeting).
- **Updating the community calendar** visible to all users.
- **Sending reminders** before deadlines or scheduled events.

This ensures deadlines are never siloed but are instead integrated into the community's shared view.

Notification Engine

Loomio's notification system works in real-time:

- **Triggers are event-based**, meaning whenever a new task is assigned, a deadline approaches, or a leave is approved, the backend queues a notification.
- **Delivery is multi-channel** – users are notified through in-app alerts and emails.
- **Redundancy is minimized** by ensuring only the most relevant notifications are sent to avoid spamming users.

Dashboards & Reporting

Processing of analytics involves:

- **Aggregating data** across tasks, contributions, attendance, and events.

- **Generating visual summaries** (graphs, charts, lists).
- **Custom filtering** to allow leaders and members to focus on specific timeframes, individuals, or events.

In essence, Loomio's major processing functionality is designed to **convert raw participation data into structured, actionable insights**, thereby enabling communities to measure productivity effectively.

5. Modules of the Proposed System

Loomio is designed in a **modular architecture** where each feature area functions as an independent yet integrated unit. This ensures scalability, maintainability, and clarity of responsibilities within the system.

5.1 User & Role Management Module

This module manages registration, authentication, and authorization. It includes:

- **User registration and login** with email and password.
- **Role assignment** (Admin, Leader, Member).
- **Secure authentication** using JWT tokens.
- **Access control** ensuring different roles see different interfaces and permissions.

5.2 Task Management Module

This core module supports:

- Task creation and assignment.
- Task status tracking.
- Deadline monitoring.
- Link sharing and comment.
- Integration with contribution points.

5.3 Contribution Assessment Module

This module quantifies member involvement by:

- Assigning points for task completion, attendance, and participation.
- Displaying contribution charts per user.

- Allowing admins to adjust weightages for fairness.
- Providing comparative analytics (e.g., top contributors).

5.4 Attendance & Leave Management Module

This module ensures transparent participation tracking by:

- Recording daily attendance.
- Processing leave applications and approvals.
- Generating attendance reports for analysis.
- Linking attendance data to contribution points.

5.5 Community Calendar Module

Centralized scheduling handled by:

- Storing community events.
- Syncing task deadlines with the calendar.
- Displaying shared calendars for all members.
- Sending event reminders and alerts.

5.6 Notification System Module

Handles the communication aspect:

- Email notifications via Nodemailer.
- Real-time in-app alerts.
- Configurable preferences for notification frequency.

5.7 Dashboard & Reporting Module

Provides a **visual representation of performance and progress** by:

- Displaying graphs for contributions, tasks, and attendance.
- Offering role-specific dashboards (Admin vs Member).
- Enabling export of reports to PDF/CSV.

5.8 Admin Panel Module

Supports administrative control through:

- User management (adding, removing, editing users).

- Monitoring system activity.
- Approving leave requests.
- Generating community-wide reports.

Together, these modules ensure that Loomio is not just a **task management tool**, but a **comprehensive community collaboration ecosystem**.

6. Business Constraints & Technical Constraints

6.1 Business Constraints

1. **Budget Limitations:** Loomio is targeted primarily at **student and volunteer communities**, many of whom operate under minimal budgets. The system must therefore leverage **free-tier deployment platforms** such as Vercel, Railway, or Render.
2. **Ease of Adoption:** Since the user base may include individuals with limited technical skills, the system must remain **intuitive and user-friendly** to minimize training needs.
3. **Recognition and Fairness:** The contribution points system must be transparent, as perceived bias could reduce adoption. Hence, all algorithms must be clearly communicated and logically sound.
4. **Non-Profit Orientation:** Loomio must avoid unnecessary overhead features common in enterprise tools. The focus should remain on **lightweight, essential features** for collaboration.

6.2 Technical Constraints

1. **Frontend Technology:** The frontend must be built in **Vite + React**, chosen for speed, responsiveness, and modern development practices.
2. **Backend Technology:** The backend must use **Express.js (Node.js)** for creating scalable APIs and handling asynchronous operations efficiently.
3. **Database Choice:** A **MySQL relational database** is mandated to ensure structured relationships between tasks, users, attendance, and contributions.
4. **Authentication:** JWT is required for secure session management, avoiding heavier alternatives like OAuth where unnecessary.

5. **Hosting:** Only **free-tier or low-cost hosting platforms** will be used during the initial deployment phase. Commercial cloud resources may be adopted later but must remain optional.
6. **Device Compatibility:** The system must be mobile-responsive to ensure accessibility across devices.
7. **Scalability Constraints:** Free-tier services may limit simultaneous connections, runtime hours, or database storage. Loomio must be optimized to work under these constraints (e.g., caching frequently accessed data).

7. User Characteristics

Loomio is designed to serve a **diverse range of users** within student communities, volunteer groups, and professional organizations. Each user category has distinct responsibilities and interaction patterns within the system. Understanding these user characteristics is critical to ensuring that the platform meets the expectations of its stakeholders.

7.1 Administrator/Community Leader

- **Role in the system:** Administrators (or leaders) are responsible for overall system management, including user onboarding, role assignments, task delegation, and approval of leave requests. They act as the supervisory layer of the system, ensuring that all community operations remain smooth and transparent.
- **Technical background:** Leaders are expected to have basic to moderate technical literacy. While they may not possess deep technical expertise, they should be comfortable navigating dashboards, creating tasks, and interpreting reports.
- **Expectations:** Administrators expect quick access to consolidated data (attendance records, contribution points, and task progress). They also expect an efficient approval workflow for task requests and leave applications.
- **Impact:** Leaders use Loomio to enforce structure, improve accountability, and ensure fair recognition of contributions within the group.

7.2 Regular Members/Participants

- **Role in the system:** Members are the backbone of the community. They perform assigned tasks, attend events, request leave, and contribute actively to the

group's goals. Their primary interaction is with the task board, calendar, and attendance modules.

- **Technical background:** Participants are assumed to have basic digital literacy (ability to log in, navigate menus, mark task status). The interface must be designed to be **simple, intuitive, and mobile-friendly**.
- **Expectations:** Members expect clear task assignments, reminders for deadlines, and fair contribution recognition. They also value transparency in attendance and contribution scoring.
- **Impact:** The success of Loomio depends heavily on member adoption. A smooth, user-friendly experience ensures continued engagement.

7.3 Observers/External Stakeholders (Optional Role)

- **Role in the system:** Observers may include faculty advisors, sponsors, or external supervisors. Their role is limited to reviewing contribution analytics, task completion rates, and attendance summaries. They do not directly participate in tasks or approvals.
- **Technical background:** Observers are expected to have minimal technical involvement. They should be able to log in and access reports without additional training.
- **Expectations:** Observers value **read-only, concise reports** and dashboards that summarize community performance without overwhelming them with unnecessary details.

Summary: Loomio's design must consider these distinct user roles, ensuring that **admins, members, and observers** each have a tailored experience. The system must remain easy to learn, fair in its assessment mechanisms, and efficient in its workflows.

8. Data Model and Description

Loomio is built on a **relational data model** implemented in MySQL. The design ensures **normalization, referential integrity, and scalability** to support growing user bases.

8.1 Core Entities and Attributes

1. **User Table**
    - Attributes: User_ID (PK), Full_Name, Email, Password_Hash, Role, Points, Join_Date.

- Purpose: Stores user credentials, roles (Admin, Leader, Member), and cumulative contribution scores.

2. **Task Table**
    - Attributes: Task_ID (PK), Title, Description, Assigned_By, Assigned_To, Deadline, Status, Comments, Submission Link.
    - Purpose: Stores tasks, including assignment details, descriptions, and deadlines.
3. **Contribution Table**
    - Attributes: Contribution_ID (PK), User_ID (FK), Task_ID (FK), Points, Feedback.
    - Purpose: Captures contribution points and feedback per user-task combination.
4. **Attendance Table**
    - Attributes: Attendance_ID (PK), User_ID (FK), Date, Status (Present/Absent).
    - Purpose: Logs user attendance for events and meetings.
5. **Leave Table**
    - Attributes: Leave_ID (PK), User_ID (FK), Start_Date, End_Date, Reason, Approval_Status.
    - Purpose: Records leave requests and their approval/rejection status.
6. **Calendar Event Table**
    - Attributes: Event_ID (PK), Title, Description, Date, Created_By.
    - Purpose: Stores all community-level events and deadlines.

8.2 Relationships

- A **User** can be assigned multiple **Tasks** (1-to-Many).
- Each **Task** may generate one or more **Contribution records** (1-to-Many).
- A **User** has many **Attendance** records (1-to-Many).
- A **User** may submit multiple **Leave requests** (1-to-Many).
- **Events** are linked to multiple users but are centrally managed by admins.

8.3 Data Integrity Rules

- User email must be unique and validated.
- Tasks cannot exist without an assigned creator.
- Contribution points are auto-generated and cannot be manually altered by members.
- Attendance records must be tied to valid calendar dates.
- Leave requests must not overlap with each other for the same user.

This structured relational model ensures **accuracy, traceability, and accountability** in community collaboration.

9. Masters Information

Masters are the **foundational datasets** that define the behavior of the system. They rarely change and are used as references for transactions.

9.1 User Master

- Stores all user accounts, including roles (Admin, Leader, Member).
- Provides authentication and access control.
- Acts as the foundation for all relational links in the system.

9.2 Role Master

- Defines the available roles in the system.
- Roles include:
    - **Administrator:** Full system privileges.
    - **Leader:** Can assign tasks, approve leaves, track contributions.
    - **Member:** Executes tasks, requests leave, tracks own contributions.

9.3 Task Master

- Maintains predefined categories of tasks (e.g., academic tasks, volunteer duties, organizational events).
- Ensures uniformity in how tasks are created and tracked.
- Supports consistency in reporting by classifying tasks under standard categories.

9.4 Event Master

- Maintains recurring events, such as monthly meetings, weekly check-ins, or annual reports.
- Ensures community members can rely on structured scheduling.

By organizing these **master datasets**, Loomio achieves **standardization, reusability, and clarity**, making the system easier to scale and manage.

10. Transaction Information

Transactions represent the **dynamic operations** performed by users in the system. Unlike master data, transactions are **frequent, mutable, and time-bound**.

10.1 Task Transactions

- Creation of a task by an admin/leader.
- Assignment of task to one or more users.
- Updates in task status (progress changes).
- Task completion confirmation and closure.

10.2 Contribution Transactions

- Automatic awarding of points upon task completion.
- Updating contribution scores when attendance or event participation is recorded.
- Admin adjustments to contribution points (in exceptional cases).

10.3 Attendance Transactions

- Daily attendance marking by leaders.
- Auto-updating attendance records based on approved leaves.
- Generating attendance reports (daily/weekly/monthly).

10.4 Leave Transactions

- Submission of leave request by a member.
- Review and approval/rejection by an admin/leader.
- Updating attendance records accordingly.

10.5 Calendar Transactions

- Scheduling new events by leaders.
- Updating/deleting events when necessary.
- Sending event reminders and deadline alerts.

Summary: Transaction data captures the **day-to-day activities** of the community. Together with master data, these transactions form the **operational backbone** of Loomio, ensuring that tasks, contributions, attendance, and events are processed in real time.

11. Data Flow Diagrams (DFD)

Data Flow Diagrams (DFDs) represent the logical flow of information within Loomio, illustrating how data moves from inputs to processes and ultimately to outputs. DFDs are particularly important for showing the **relationship between modules** and how user interactions translate into system operations.

11.1 Context Level DFD (Level 0)

- **Actors:**
  - Administrator/Leader
  - Member
  - External Observer (optional)
- **System Boundary:** Loomio
- **Major Flows:**
  - Admin inputs (task creation, attendance, events) → System processes → Outputs (dashboards, notifications, reports).
  - Member inputs (task updates, leave requests, attendance confirmations) → System processes → Outputs (personal dashboards, contribution summaries).
  - Observer inputs (login requests) → Read-only views of analytics.

At this level, Loomio is represented as a **single black box**, with interactions flowing between users and the system.

11.2 Level 1 DFD (Expanded Processes)

The main system functions expand into the following sub-processes:

1. **User & Role Management**

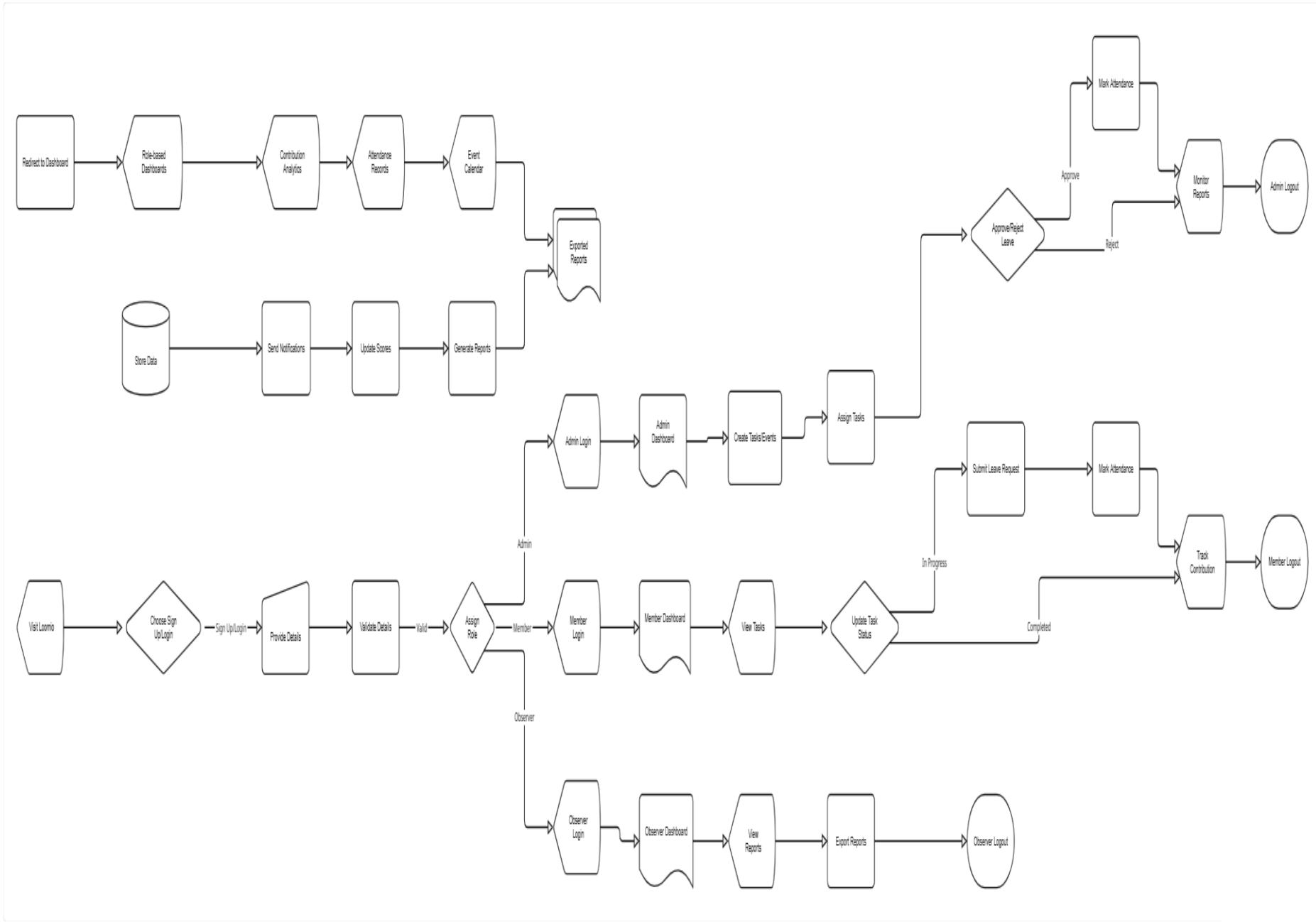
- Input: Registration details, login credentials.
- Output: User authentication, access rights.

2. **Task Management**
    - Input: Task creation and updates.
    - Output: Task lists, progress reports.
3. **Contribution Assessment**
    - Input: Task completions, attendance, events.
    - Output: Contribution points, analytics.
4. **Attendance & Leave**
    - Input: Attendance marking, leave requests.
    - Output: Attendance records, approved/rejected leave.
5. **Event Scheduling**
    - Input: Event creation and updates.
    - Output: Shared calendar, reminders.
6. **Reporting & Dashboards**
    - Input: Raw system data.
    - Output: Graphs, summaries, exported reports.

11.3 Data Stores

- **User Database**: Holds all registered users and roles.
- **Task Database**: Stores task records and progress logs.
- **Contribution Database**: Maintains contribution points and analytics.
- **Attendance/Leave Database**: Logs presence, absences, and leave requests.
- **Calendar Database**: Maintains all scheduled events.

Through these data flows, Loomio ensures that **inputs are consistently transformed into structured outputs**, enabling transparency and accountability.


Visit Loomio
Choose Sign Up/Login
Sign Up/Login
Provide Details
Validate Details
Valid
Assign Role
Admin
Member
Observer
Admin Login
Admin Dashboard
Create Tasks/Events
Assign Tasks
Approve/Reject Leave
Approve
Reject
Mark Attendance
Monitor Reports
Admin Logout
Exported Reports
Store Data
Send Notifications
Update Scores
Generate Reports
Member Login
Member Dashboard
View Tasks
Update Task Status
In Progress
Completed
Submit Leave Request
Mark Attendance
Task Contribution
Member Logout
Observer Login
Observer Dashboard
View Reports
Export Reports
Observer Logout
Redirect to Dashboard
Role-based Dashboards
Contribution Analytics
Attendance Records
Event Calendar

12. Specific Requirements

This section details the specific technical and user-facing requirements for Loomio.

12.1 User Interface Requirements

- The UI shall be **responsive** across devices (desktop, tablet, mobile).
- The UI shall provide **role-based dashboards**:
    - Admin Dashboard: Task assignment, attendance approval, analytics.
    - Member Dashboard: Task list, personal attendance, contribution summary.
    - Observer Dashboard: Read-only community performance reports.
- The UI shall present **task boards** with clear visual indicators (color-coded for Pending, In Progress, Completed, Overdue).
- The calendar interface shall allow users to view events in **monthly, weekly, and daily formats**.
- All dashboards shall include **graphical analytics (bar charts, pie charts)** to visualize contributions and attendance.
- Notifications shall appear as **in-app alerts and email messages**.

12.2 Hardware and System Software Requirements

Minimum Hardware Requirements:

- **Server-Side:**
    - Processor: Dual-core 2.0 GHz or higher.
    - RAM: 4 GB.
    - Storage: 20 GB.
- **Client-Side (User Devices):**
    - Any device capable of running a modern browser.
    - Recommended: 2 GB RAM, dual-core processor.

System Software Requirements:

- **Backend Server OS:** Linux (Ubuntu) or Windows Server.
- **Frontend:** Runs on any modern browser (Chrome, Firefox, Edge, Safari).

- **Database:** MySQL 8.0+
- **Development Environment:** Node.js (Express.js), Vite, React.

12.3 Network/Communication Interface

- The system shall operate over **HTTP/HTTPS protocols**.
- Client-server communication shall use **RESTful APIs**.
- Notifications shall use **SMTP email services**.
- System must support **minimum 10 concurrent users** on free-tier hosting, scalable to 100+ users with optimization.
- Latency between user input and response shall not exceed **2 seconds** under normal loads.

12.4 Software Interface Requirements

- **Database Interface:** The backend shall interface with MySQL using mysql2 or sequelize.
- **Authentication Interface:** JWT shall be used to manage user sessions securely.
- **Email Interface:** Nodemailer shall be integrated with external SMTP services (e.g., Gmail, Mailgun).
- **Frontend-Backend Interface:** All communication shall be JSON-based over REST APIs.
- **Export Interfaces:** Reports shall be exportable to **CSV and PDF formats**.

13. Performance Requirements

Performance requirements define how Loomio must behave under different conditions to ensure **usability, reliability, and scalability**.

13.1 Response Time

- Login and authentication shall complete within **2 seconds**.
- Task creation and assignment shall complete within **3 seconds**.
- Dashboards must load within **5 seconds** for communities of up to 500 users.

13.2 Throughput

- The system shall handle at least **50 task updates per minute** under peak usage.
- Attendance marking for a group of 100 users shall complete within **1 minute**.
- Event scheduling shall support **simultaneous updates by multiple leaders** without conflict.

13.3 Availability & Reliability

- The system shall be available **99% of the time** under standard free-tier hosting.
- Daily backups of the MySQL database shall be automated to prevent data loss.
- Error handling shall provide **meaningful messages** (e.g., “Task deadline missing” instead of generic errors).

13.4 Security Performance

- All sensitive data (passwords, authentication tokens) shall be encrypted using industry-standard hashing.
- JWT tokens shall expire after a defined session period, reducing risks of misuse.
- Access shall be role-restricted to prevent unauthorized task updates or data access.

13.5 Scalability

- Loomio shall initially support **small communities (20–100 users)** with seamless performance.
- The architecture shall allow scaling to **1000+ users** with database indexing, query optimization, and possible migration to cloud-hosted paid tiers.

13.6 Maintainability

- Code shall follow **modular design** principles to allow independent updates to task, attendance, or calendar modules.
- Clear API documentation shall be maintained for integration with third-party tools in the future.

14. Future Enhancements

14.1 Desktop Application:

- Develop a **cross-platform desktop app** for **Windows, Mac, and Linux**. This would allow users to access Loomio even when they are offline or prefer a native desktop experience.

  Features could include:

  - **Offline task management:** Syncing with the server once reconnected.
  - **Push notifications** for task updates, deadlines, and messages, integrated directly into the OS.
  - **Drag-and-drop task management** for smoother workflows.
  - **Integration with system calendars** for event and deadline reminders.

14.2 Mobile Application:

- Launch **native mobile apps** for **iOS and Android**, providing users with an optimized mobile experience for on-the-go task management.

  Features could include:

  - **Push notifications** for deadlines, task updates, and event reminders.
  - **Mobile-friendly task management** with a simple, intuitive UI for quick interactions.
  - **Barcode scanning** for event check-ins or task submissions (e.g., scan a code when a task is completed).

14.3 AI-Powered Task Suggestions:

- Introduce **AI-powered task recommendations** based on past contributions, member skills, and team needs. This can help leaders delegate tasks more effectively by suggesting the right people for the right tasks.

  Features:

  - **Skill-based task matching:** AI analyzes past performance and suggests tasks to users based on their strengths.
  - **Workload balancing:** AI suggests tasks to ensure no one is overloaded.