

### Motivation

ACM SIGSOFT publishes a bi-monthly hybrid Newsletter called Software Engineering Notes (SEN). The Newsletter is manually assembled by the editor, who receives various contributions from established column editors and other members of the software engineering industry who submit documents in various forms and formats as part of a document workflow.

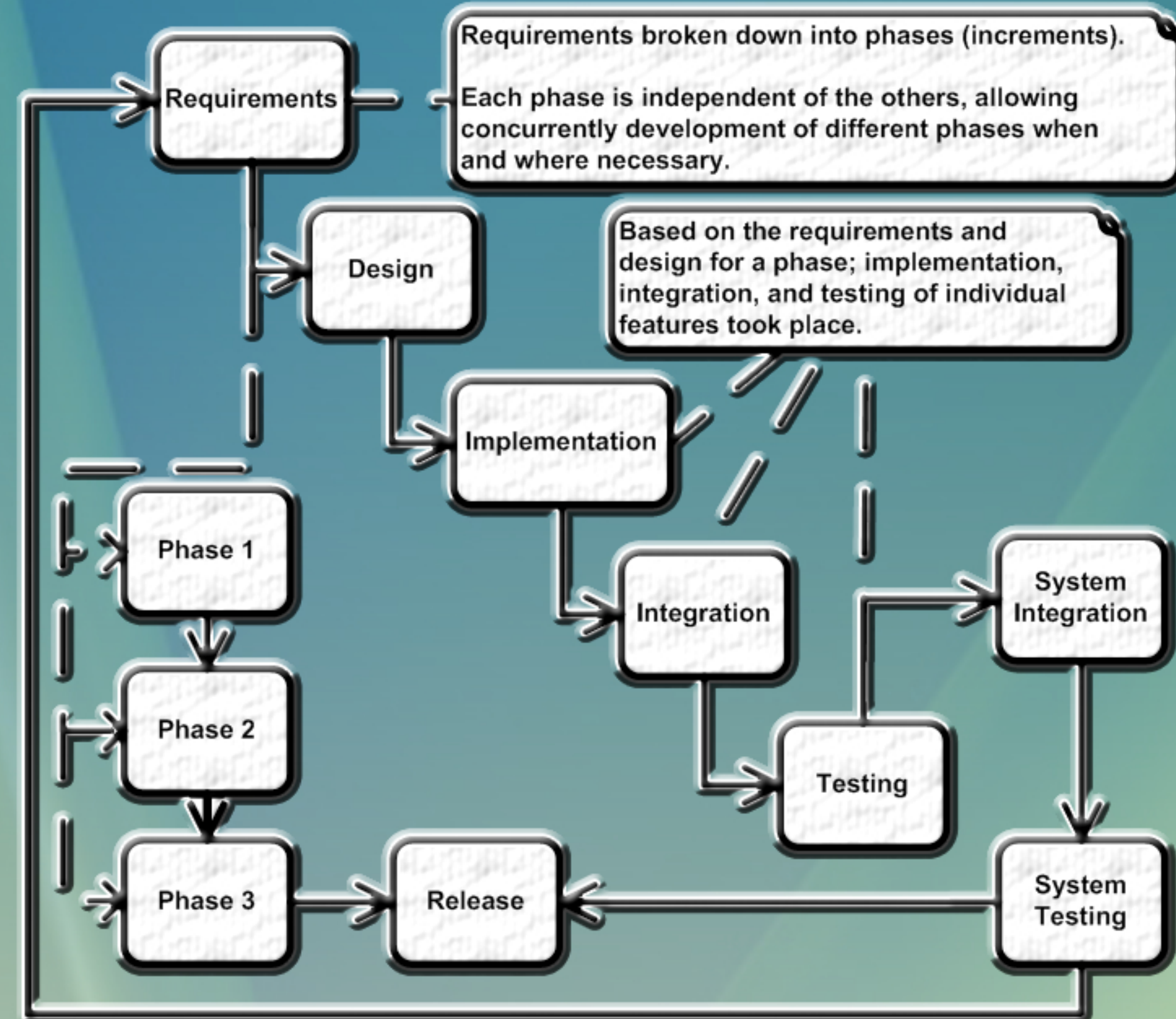
Manual assembly of the Newsletter is a daunting task.

### Background

The system has two main components:  
 • Submission  
 • Composition

The application acts as a content management system and allows users of the system to submit documents in any format. These documents will be accompanied by metadata, which is necessary to automate the construction of the newsletter. The system performs automatic composition of the various tables required for the newsletter. Once composed, these tables will be combined into digital format for construction of the remaining parts of the newsletter.

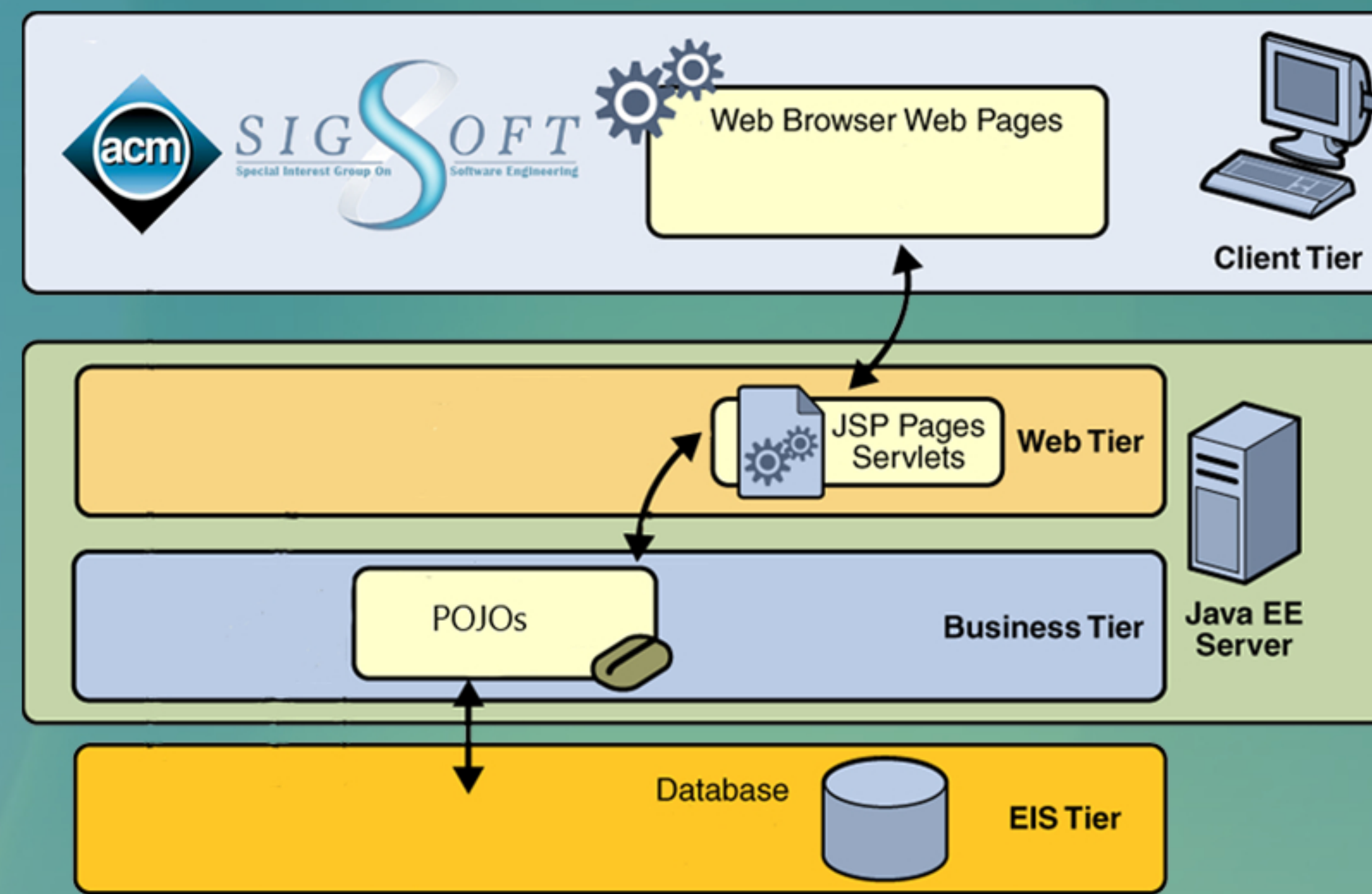
### Process Methodology



**Iterative Process**  
 The use of iterative development provided a platform for our team to continually revise and rework the system's major features to meet our sponsor's evolving requirements and apply feedback gathered through incremental delivery.

**Incremental Phased Delivery**  
 Separation of the requirements into incremental phases allowed our team to prioritize requirements, work on different phases concurrently, and gauge our progress. It also allowed our sponsor to view works-in-progress and provide us with valuable feedback.

- Phase 1**  
 Focused on user management, file submission, data collection, and manual notifications.
- Phase 2**  
 Focused on administrative functionality, user roles, and event-based notifications.
- Phase 3**  
 Focused on newsletter generation, logging, and time-based notifications.

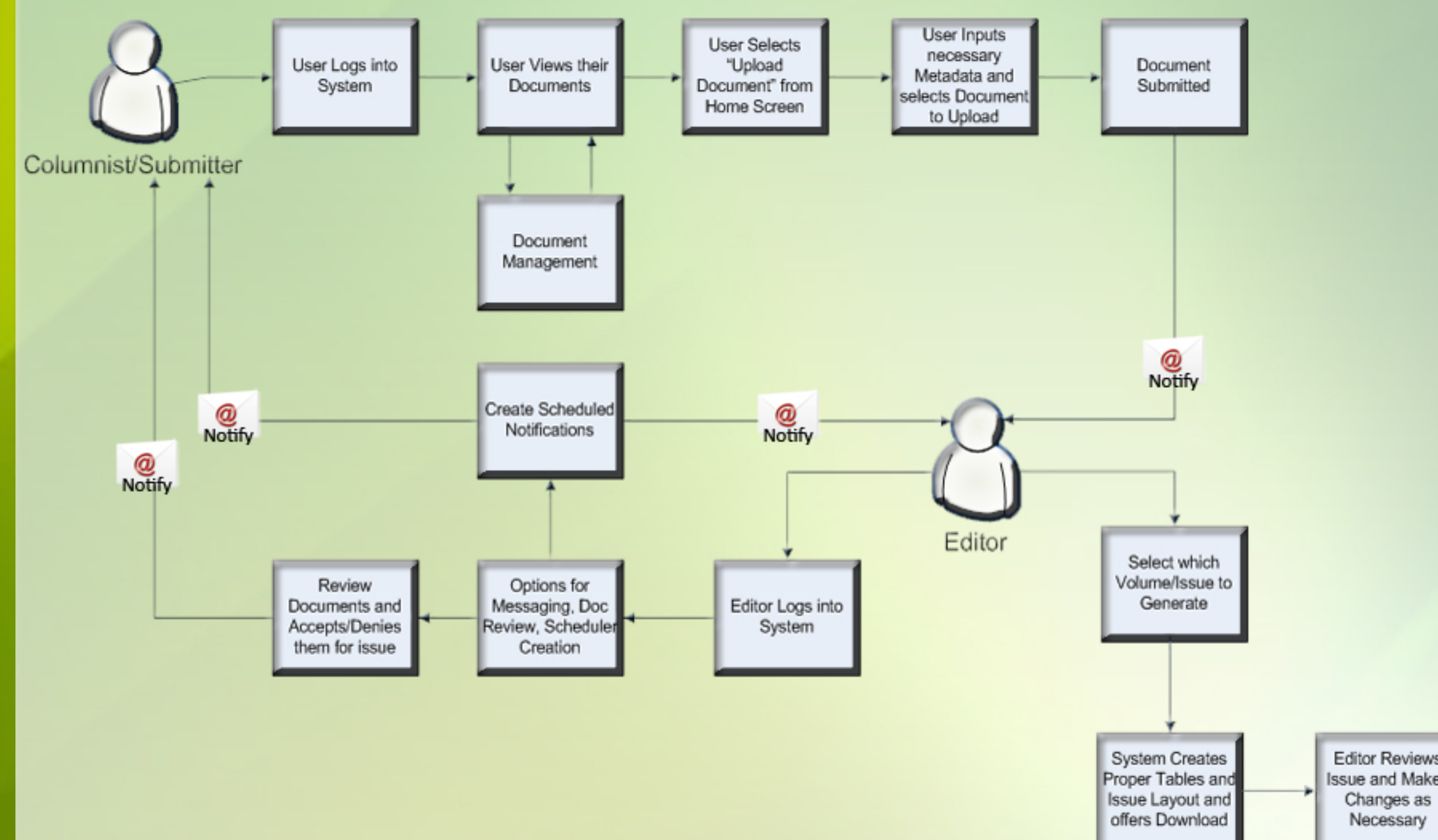


### Architecture

**User Interface & Logic (Model-View-Controller)**  
 Provides a separation of concerns between the three major parts of the system: the user interface, the business logic, and the data itself. The user-interface acts as the view and accepts user input. These inputs are then handled by servlets which act as the controller and communicate with the application layer, which acts as the system's model.

**Newsletter Creation (Pipe & Filter)**  
 Provides a method for separating the task of generating the tables within a newsletter into separate steps. This allows each step of table generation to be developed separately and then integrated at the end as a sequence of steps resulting in the generation of a table. Following this pattern also improves maintainability and testability, as maintenance and testing can be performed on each step individually. This ensures each step works properly before it is integrated with the other steps.

**Notifications & Event Logging (Event-driven)**  
 Provides notifications to one or more relevant users should an event occur within the system. Editors also have the power to send custom notifications through the user-interface. Notifications will be triggered either in response to some action or in response to the real world's time and date. All major events are logged within the system.



### Technologies Used

- Apache Tomcat 6.x
- MySQL 5.0
- OpenOffice UNO
- Java 6
- Internet Explorer 7
- Firefox 3.x
- jQuery 1.2.6
- JavaScript
- TinyMCE

### Features

- User Registration
- User Authentication
- User Management
- Document Submission
- Metadata Submission
- Content Management
- Manual Notifications
- Event-based Notifications
- Scheduled Notifications
- Metadata Extraction
- Administrative Options
- Frontmatter Composition
- Backmatter Composition
- Logging

