Team Report

Systems Design And Security PROject

Urszula Talalaj | Ting Guo | Julia Derebecka | Huiqiang Liu

06.12.2019

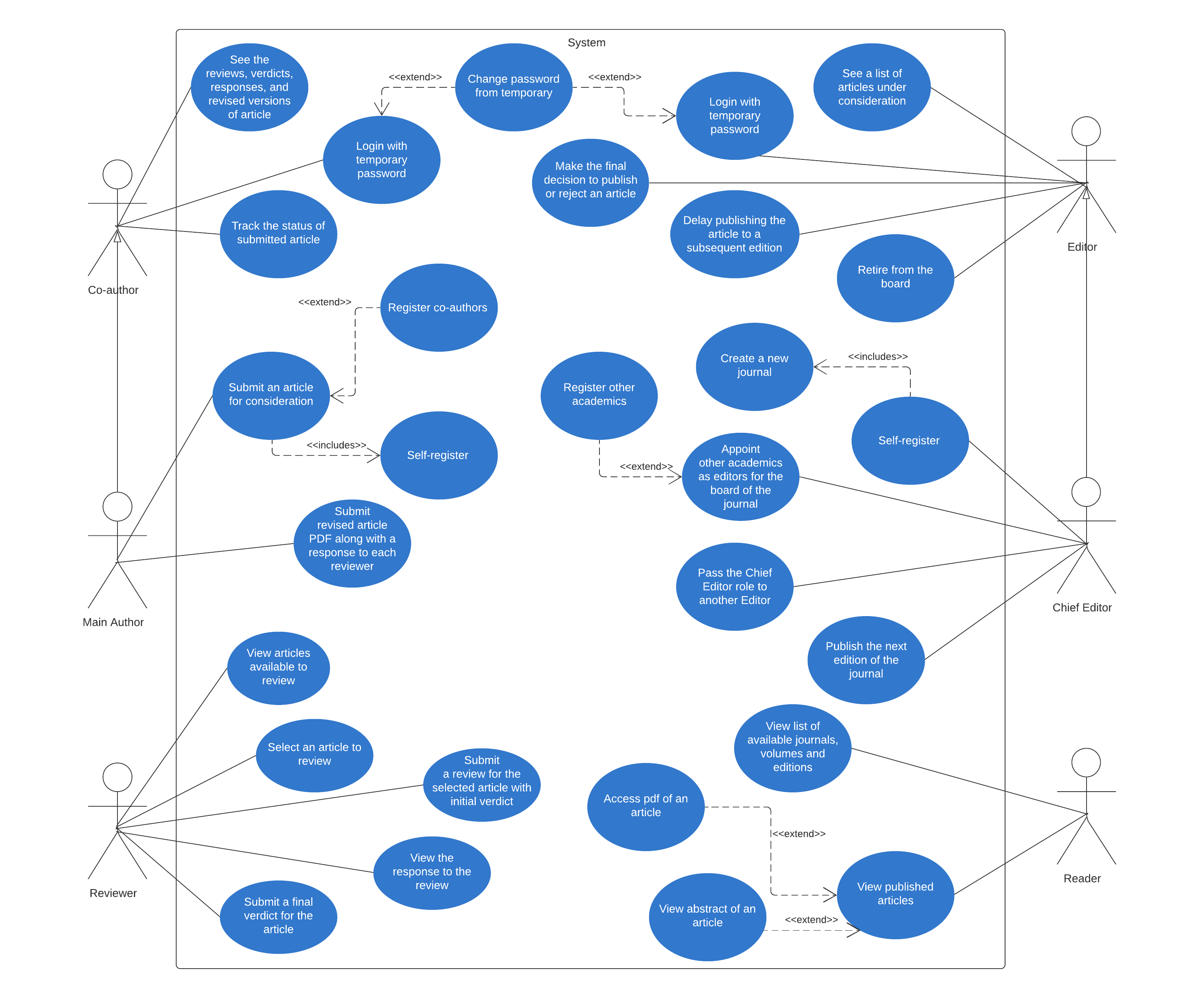
# Introduction

The objective of the project was to create a software system for an academic publisher. The system would be used for managing the submission, reviewing and publishing of academic articles for the journals operated by the publisher. Users of the publication system include readers, editors, authors and reviewers.

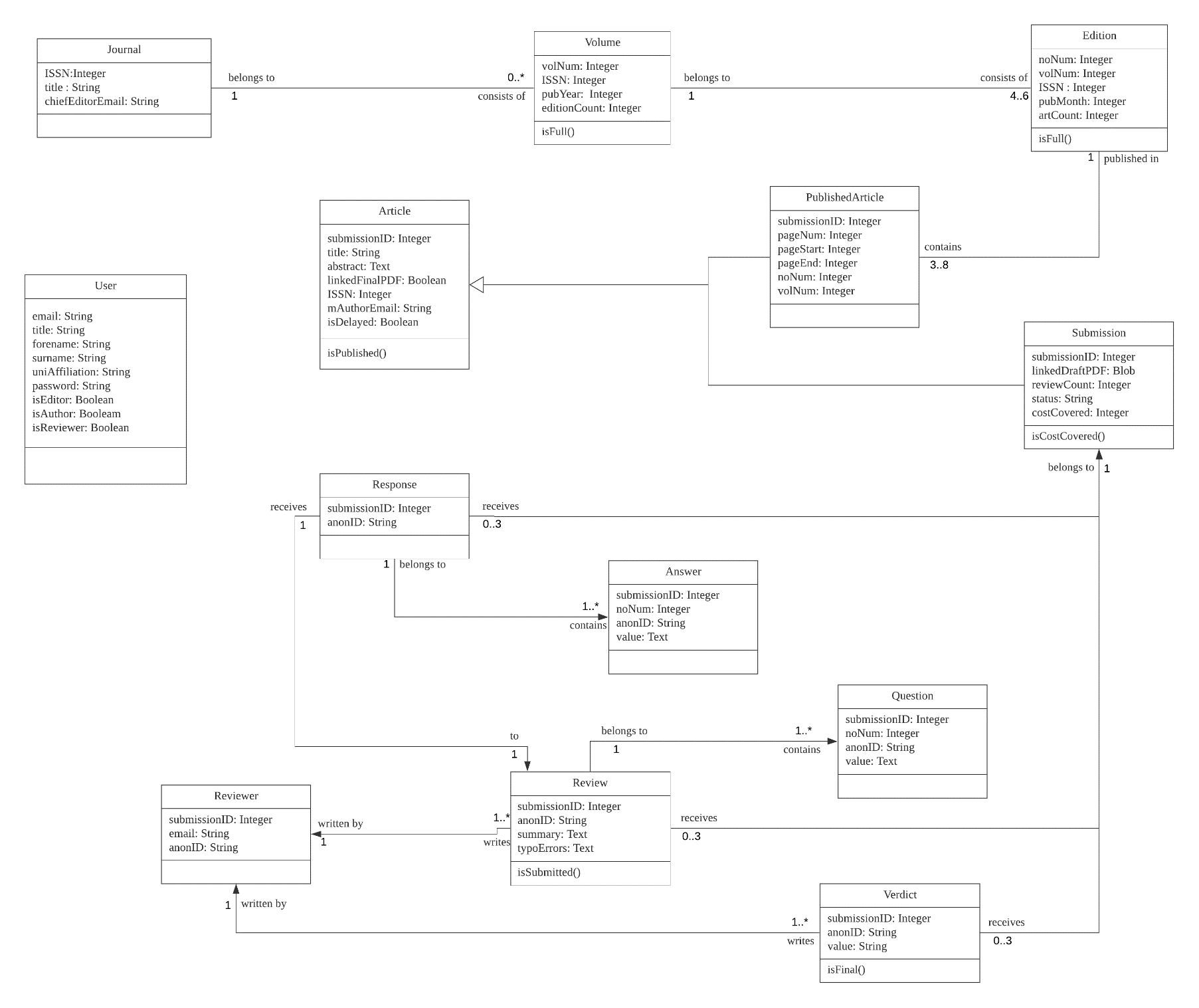
##### Interpretation of the requirements

* Adding a new journal is combined with a chief editor self-registration.
* The chief editor decides about the contents of the next edition of the journal. He can choose from all the articles accepted by all the editors (himself included) and add them to the next edition or delay publishing to a subsequent edition.
* Chief editor does not create a new volume manually. The volume for a given journal is created automatically when a new edition is published only if there is no current volume in which the edition could be published.
* Conflict of interest are detected based on the university affiliation of editors, reviewers and authors which is provided by them during the registration or later if their account has been created by someone else. The university affiliation is a name of the institution which the stakeholder has a formal connection with, and it should be supplied correctly and precisely as it is.
* Authors can only respond to the reviews when all three of them had been submitted.
* Answers must be added in such order, so they correspond to the questions.
* Reviewers can review three articles for each submission they have made. When they select an article to review, the counter of remaining articles that need to be reviewed in order to cover the cost of publishing **one of their submissions** decreases. This counter is shared between all authors of the submission. If authors have more than one submission for which they need to cover the cost of publishing, then the counter displayed for them in the reviewer interface is a sum of all their submissions’ counters.
* Reviewers lose their privileges when the last of the three reviewers of the submission submits their final verdict and their counter of remaining articles that need to be reviewed is equal to zero.
* Authors lose their privileges when their only remaining submission is accepted or rejected by an editor.
* Users can log in to only one of their roles at a time.
* Once logged in, users can change their password or update their details which requires supplying the password again.

# UML use case diagram



# UML class diagram of the initial information model



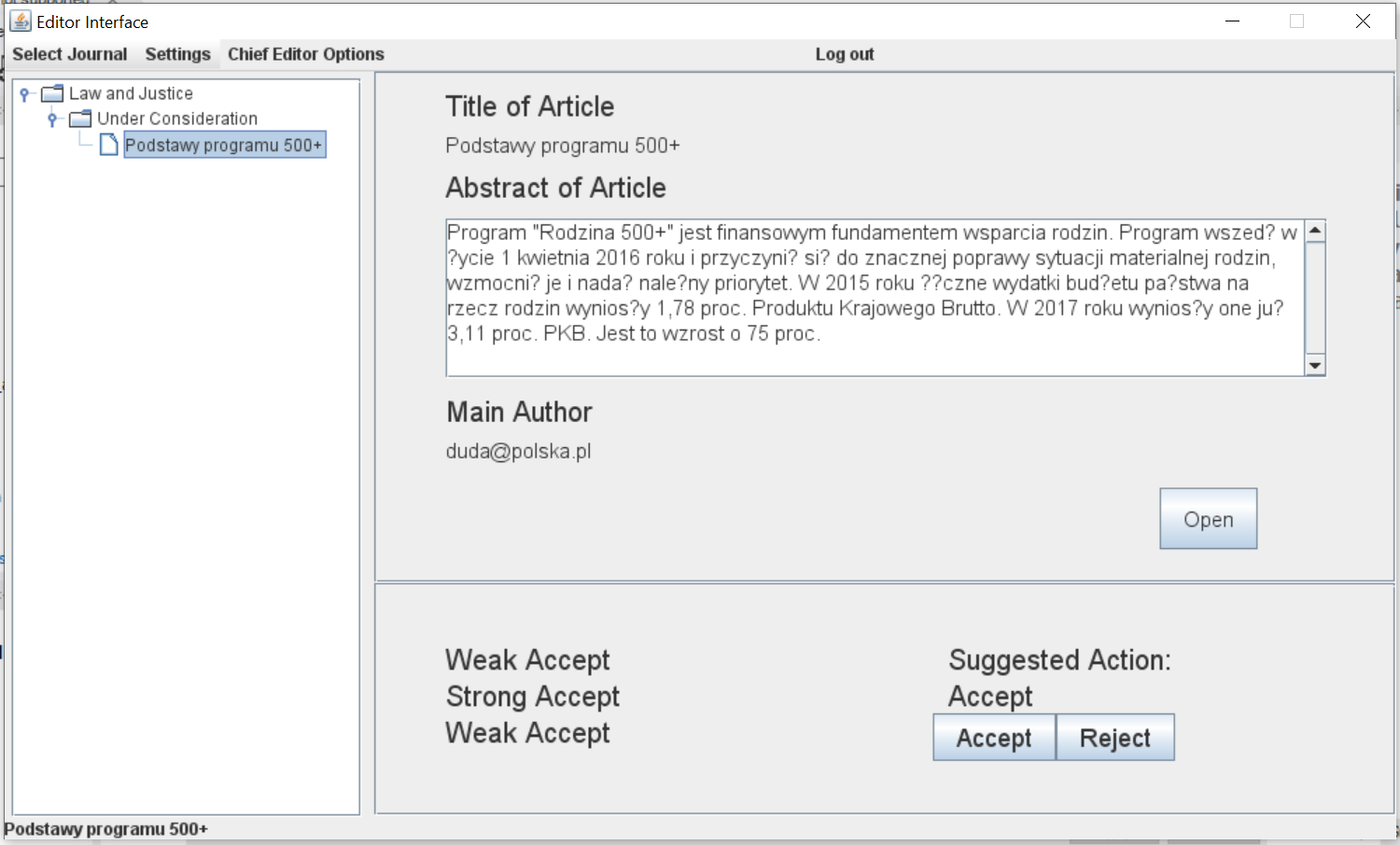
# UML class diagram of the normalised database model

*NOTE:* We are aware of the 1 to 1 relationship of Review and Response and we understand that it is not correct. The problem arose at the very end of the project and it was too late to change the database design at that point. When designing the database, we intended to have multiple responses for one review which would make this relationship one to many.

# UML state machine diagram

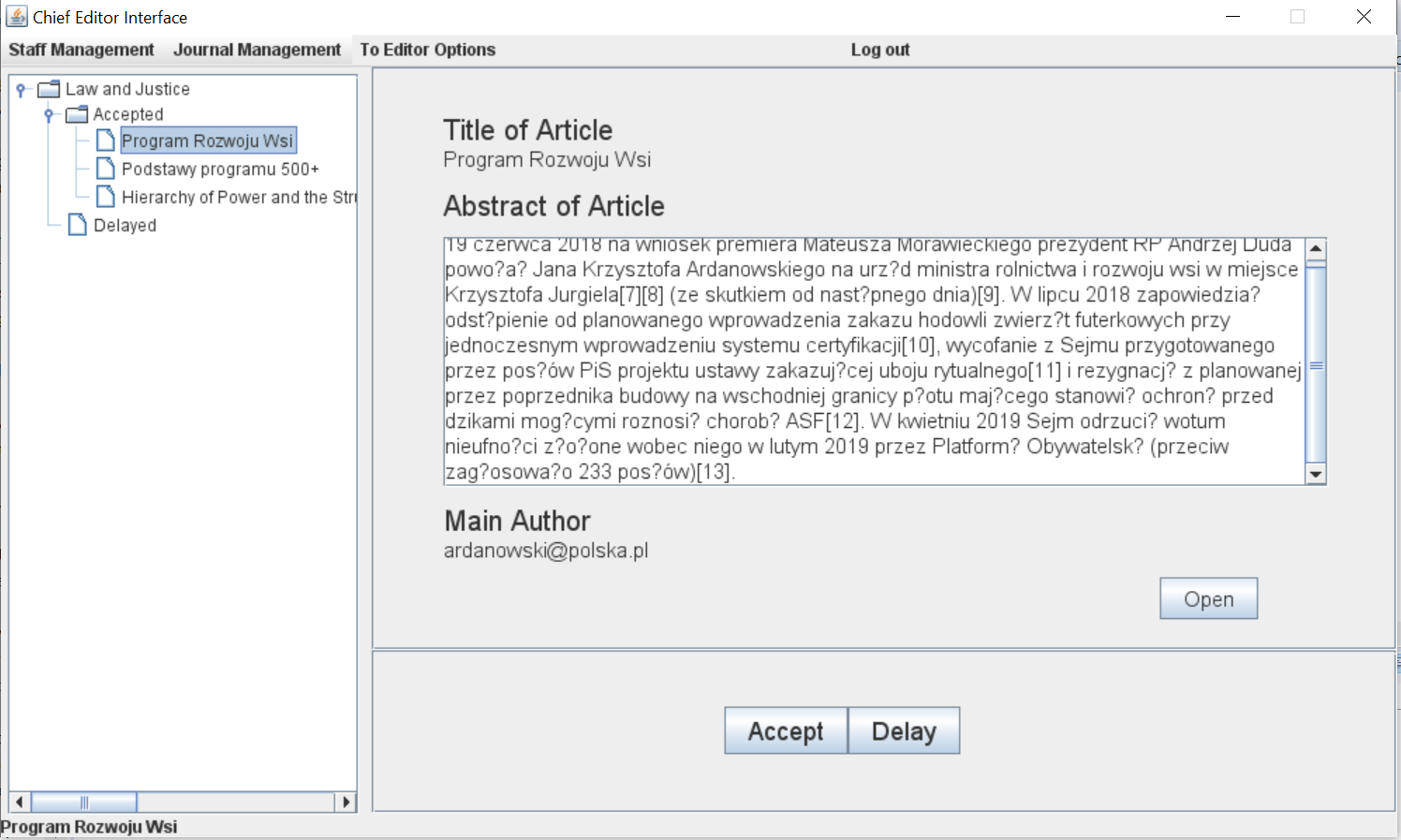
# Best aspects of our system

#### Editor Interface



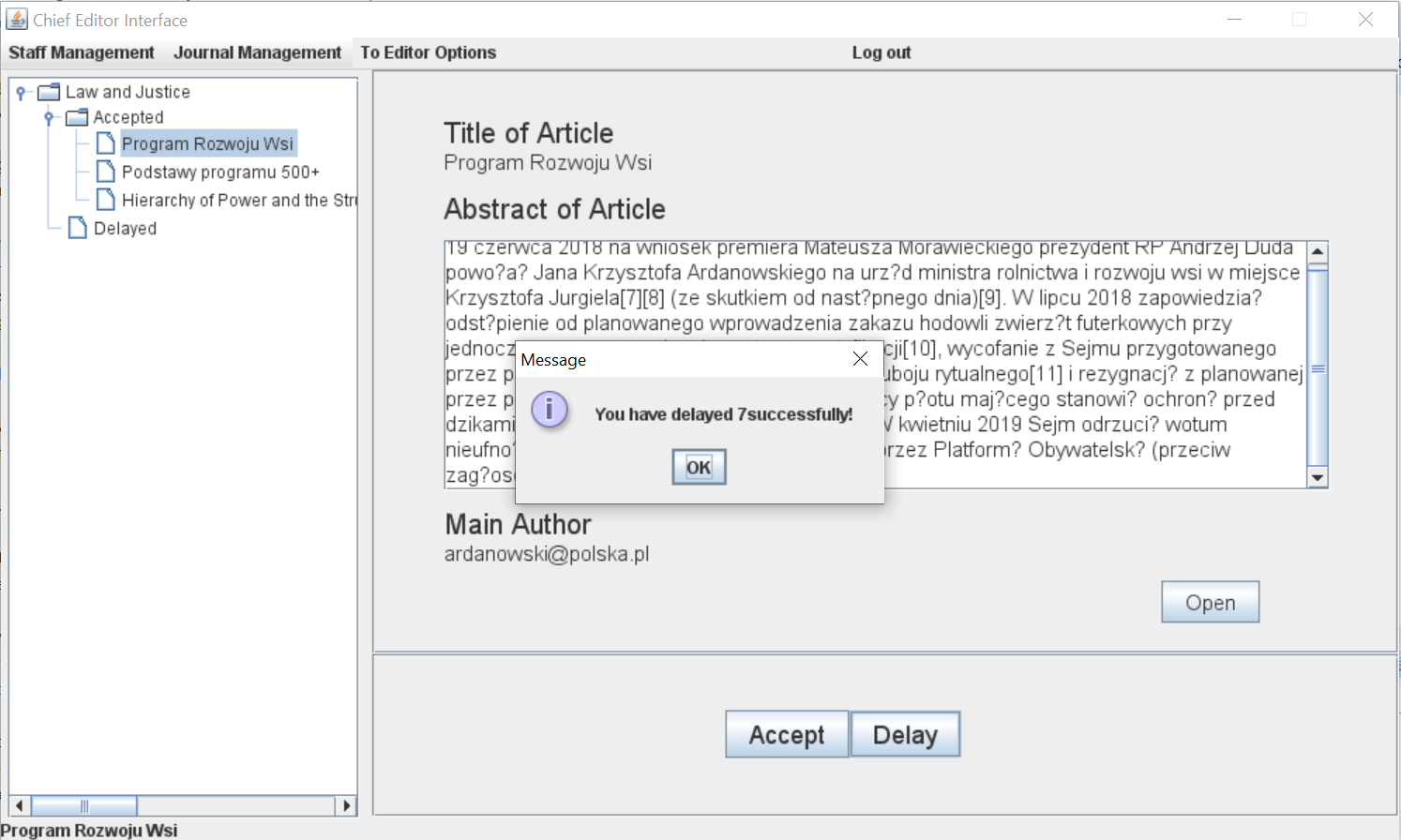
We think that the Editor Interface is one of the best interfaces in our system. Its functionality involves switching between editor’s journals, browsing the articles under consideration, viewing article title, abstract, main author, list of final verdicts and suggested action to be taken by the editor. Article final PDF can be viewed by clicking the “Open” button under article details.

#### Chief Editor Interface

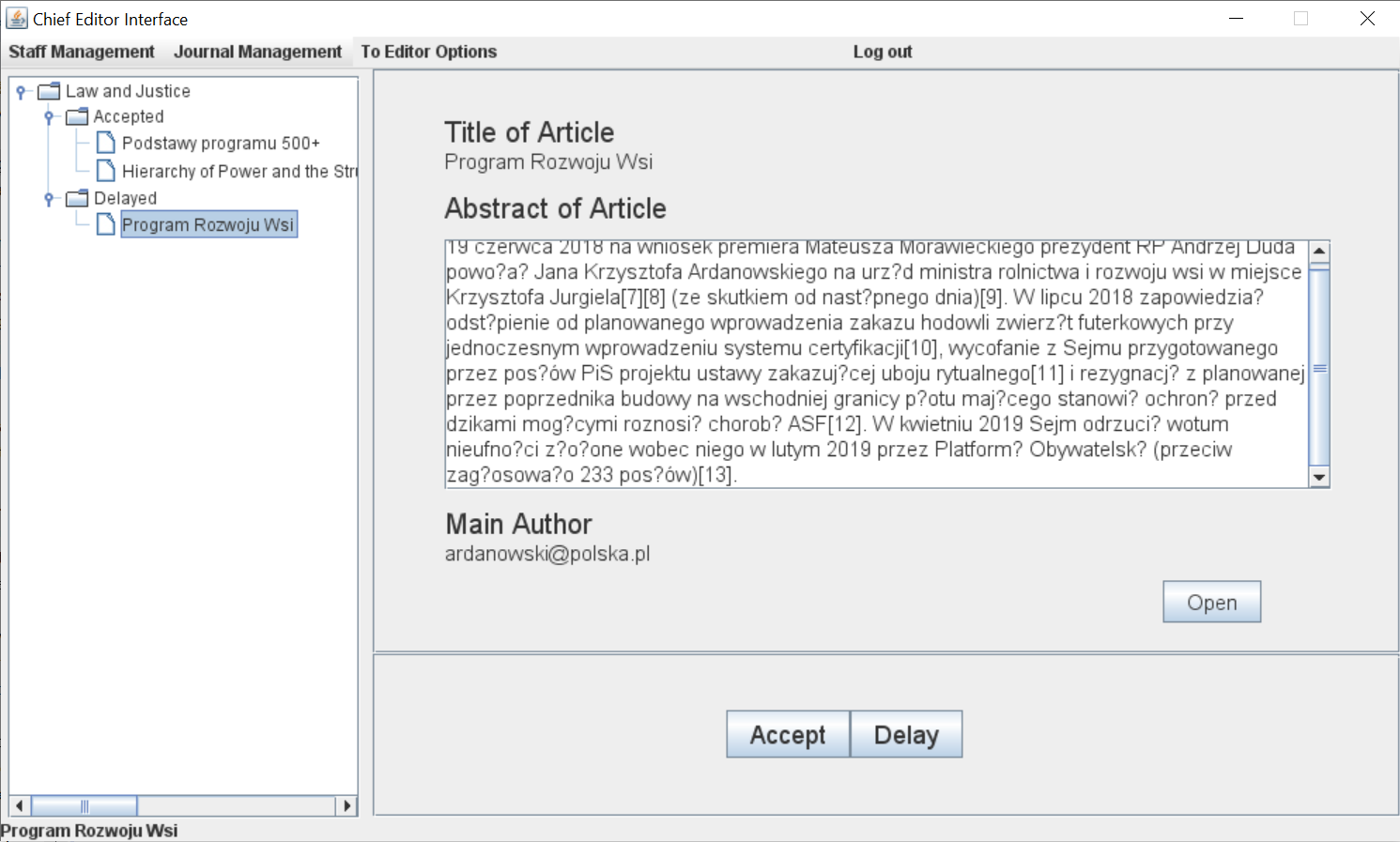


The Chief Editor Interface functionality involves switching between editor’s journals and browsing the accepted and delayed articles. The chief editor can register a new editor or appoint one already existing, pass the chief editor role to a chosen editor or retire as a chief editor with automatic appointment of a new chief editor.

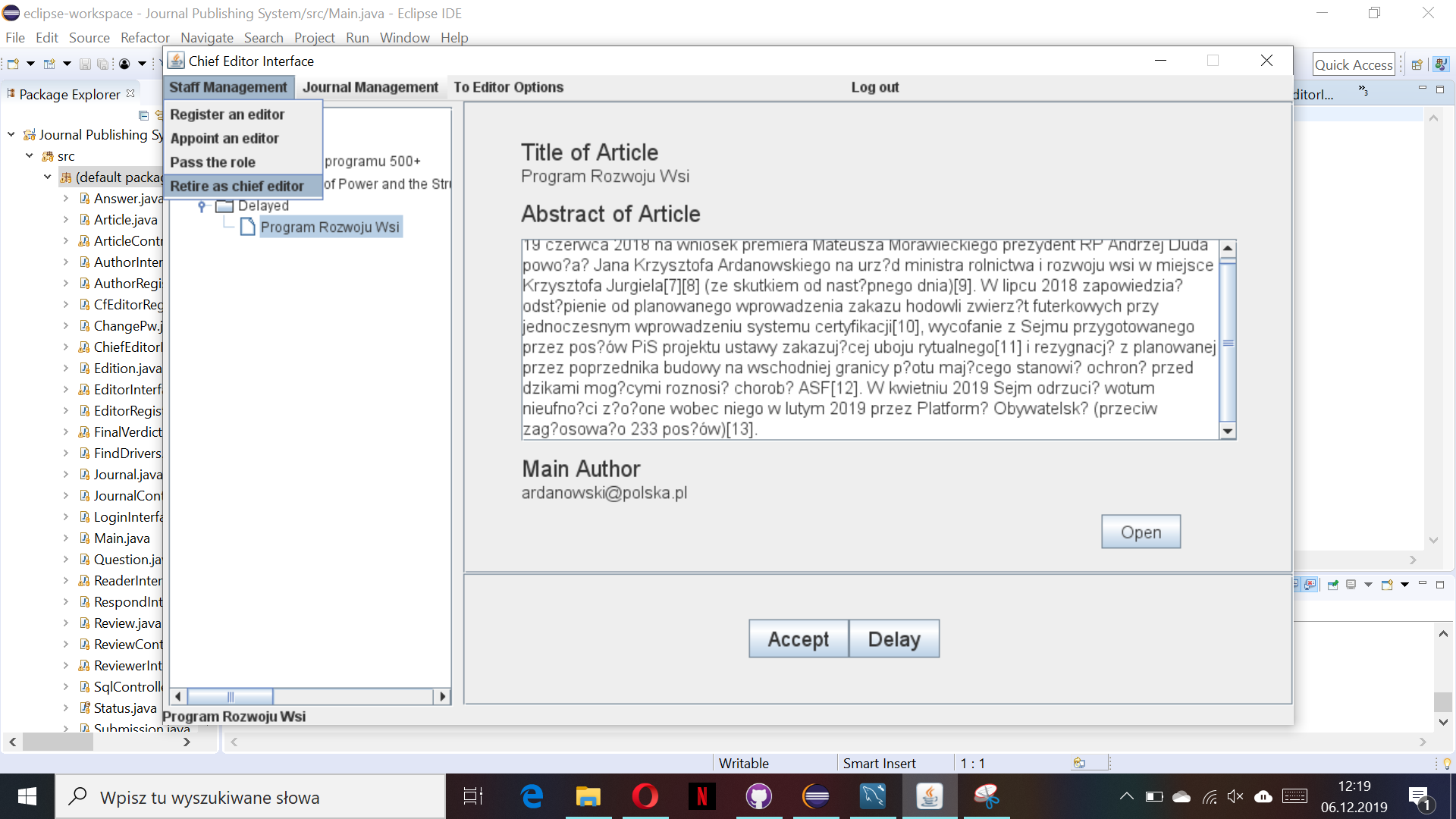
After clicking “Delay” button:



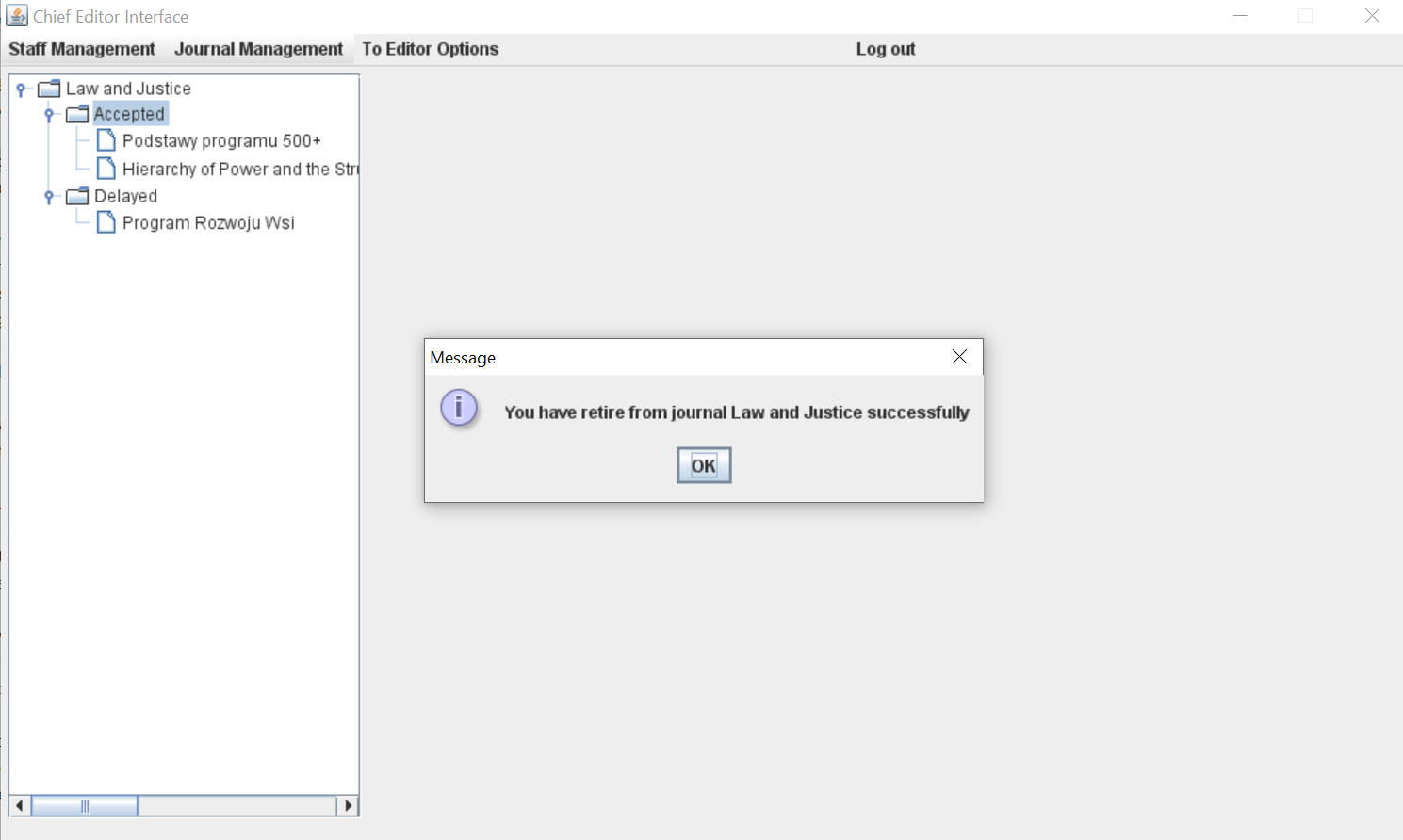
Article publication delayed:



Retire as a chief editor:



Retirement successful:



# Security features

* All registration forms have email validation.
* All registration forms have password strength check – password must be at least 8 characters, a digit must occur at least once, a lower-case letter must occur at least once, an upper-case letter must occur at least once, and spaces are not allowed.
* Passwords are securely stored in the database in hashed form using SHA2. To verify if password provided during the login is correct it is hashed, and the value is compared with the one stored in the database.
* All database queries which include user input are executed using parameterized queries (Prepared Statement) and corresponding setter methods. This can prevent most of the SQL Injection attacks.

# Measures of the effort

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Actual Tasks** | **Points** |
| Urszula Talalaj | Database classes and methods, meeting minutes, UML diagrams, team report | 32 |
| Ting Guo | Interface classes and methods | 31 |
| Julia Derebecka | Database classes and methods, UML diagrams | 30 |
| Huiqiang Liu | Interface classes and methods | 7 |