Software Requirement Specifications (SRS)

With an Emphasis on Use Cases

Most of the slides are adapted from the previous quarter's recitation and an existing SRS document

Use Cases

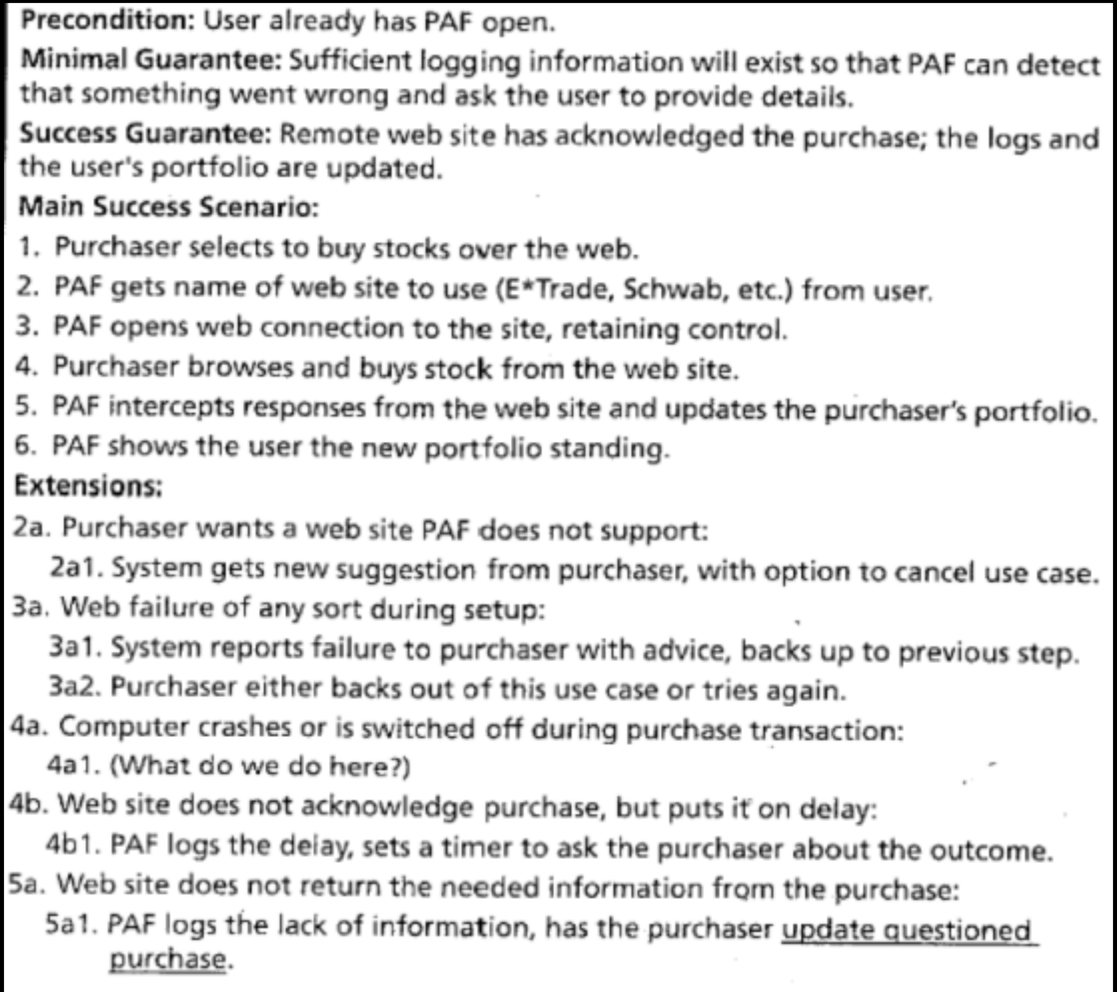
A written description of the user's interaction with the software product to accomplish a goal

- Accurately describes the what system must do
- Documentation of **functionality** not design
  - Remember **‘what’** vs. ‘how’

A Few Use Case Clarifications…

- **Primary Actor:** the person interacting in the use case
- **Level:** granularity, big picture vs. small feature
- **Minimal/Success "Guarantee"**
  - Guarantee == Post-condition
  - Minimal/Success == worst case/best case
- **Main Success Scenario**
  - Describes how user uses system to successfully complete a task
- **Extensions**
  - Alternate paths in success scenario (usually failure cases)
  - Should maintain minimal guarantee

Personal Advisors Finance Package



Precondition: User already has PAF open.

Minimal Guarantee: Sufficient logging information will exist so that PAF can detect that something went wrong and ask the user to provide details.

Success Guarantee: Remote web site has acknowledged the purchase; the logs and the user's portfolio are updated.

Main Success Scenario:

1. Purchaser selects to buy stocks over the web.
2. PAF gets name of web site to use (E*Trade, Schwab, etc.) from user.
3. PAF opens web connection to the site, retaining control.
4. Purchaser browses and buys stock from the web site.
5. PAF intercepts responses from the web site and updates the purchaser's portfolio.
6. PAF shows the user the new portfolio standing.

Extensions:

2a. Purchaser wants a web site PAF does not support:
   2a1. System gets new suggestion from purchaser, with option to cancel use case.
3a. Web failure of any sort during setup:
   3a1. System reports failure to purchaser with advice, backs up to previous step.
   3a2. Purchaser either backs out of this use case or tries again.
4a. Computer crashes or is switched off during purchase transaction:
   4a1. (What do we do here?)
4b. Web site does not acknowledge purchase, but puts it on delay:
   4b1. PAF logs the delay, sets a timer to ask the purchaser about the outcome.
5a. Web site does not return the needed information from the purchase:
   5a1. PAF logs the lack of information, has the purchaser update questioned purchase.

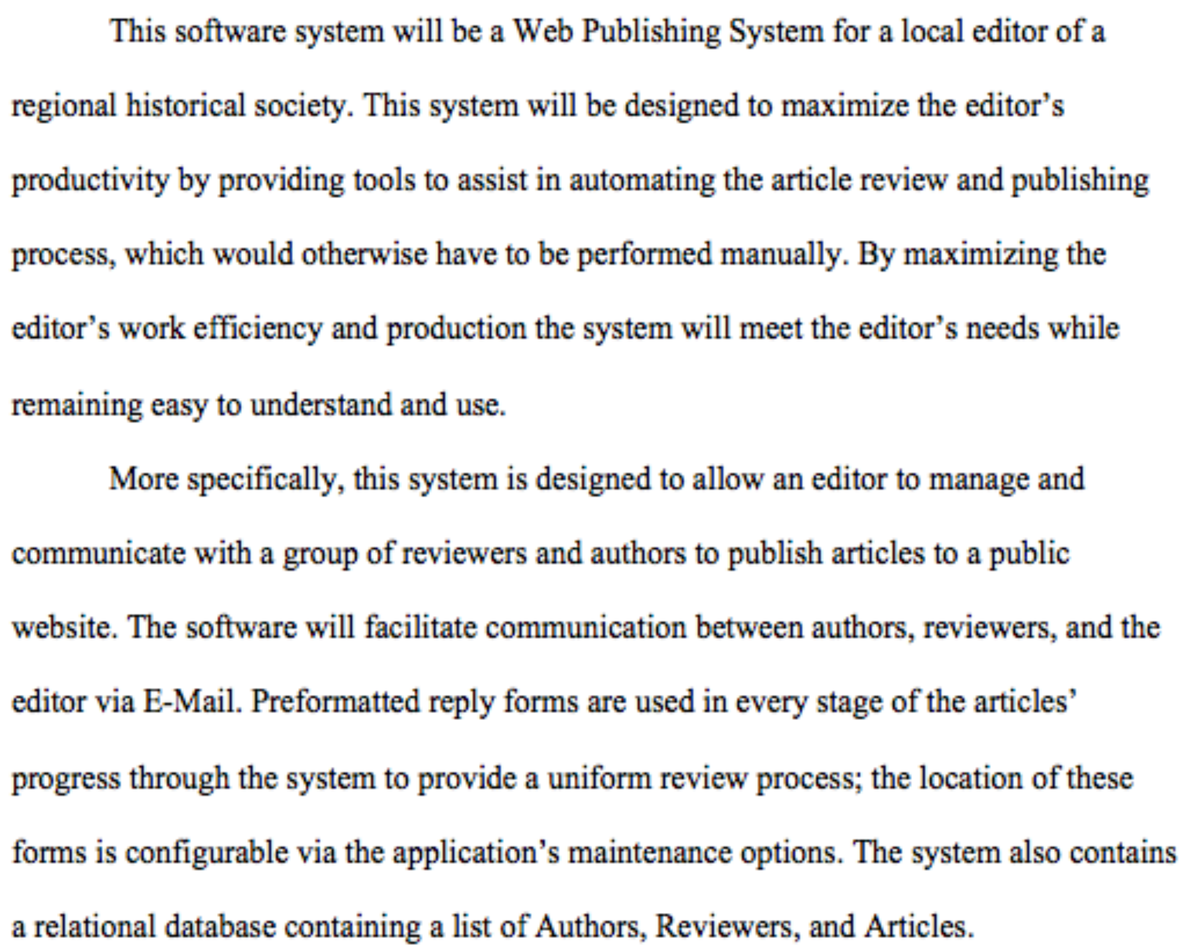
SRS Definition

> “A software requirements specification (SRS) is a complete description of the behavior of a system to be developed.”
>
> Wikipedia

SRS Outline

- Introduction: Purpose, scope and overview
- Overall Description
  - System environment
- Functional Reqs: **"What"** your software does
  - Use case scenarios
- User Characteristics: How will different actors interact with your software
- Non-functional Requirements: Imposed by the employer & users
  - Quality standards, performance requirements, etc

Example: Web Publishing System



This software system will be a Web Publishing System for a local editor of a regional historical society. This system will be designed to maximize the editor's productivity by providing tools to assist in automating the article review and publishing process, which would otherwise have to be performed manually. By maximizing the editor's work efficiency and production the system will meet the editor's needs while remaining easy to understand and use.

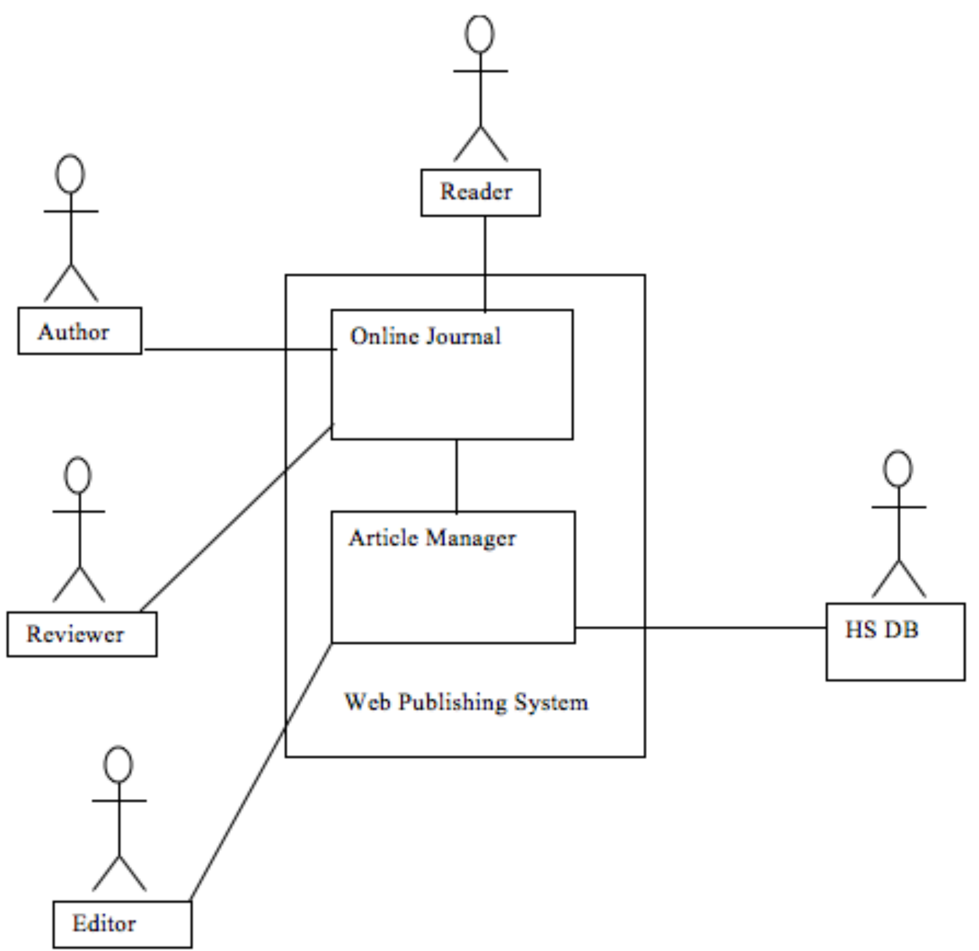
More specifically, this system is designed to allow an editor to manage and communicate with a group of reviewers and authors to publish articles to a public website. The software will facilitate communication between authors, reviewers, and the editor via E-Mail. Preformatted reply forms are used in every stage of the articles' progress through the system to provide a uniform review process; the location of these forms is configurable via the application's maintenance options. The system also contains a relational database containing a list of Authors, Reviewers, and Articles.

Example from: www.courses.utep.edu/portals/870/S04-F04%20SRS%20v0.91.doc

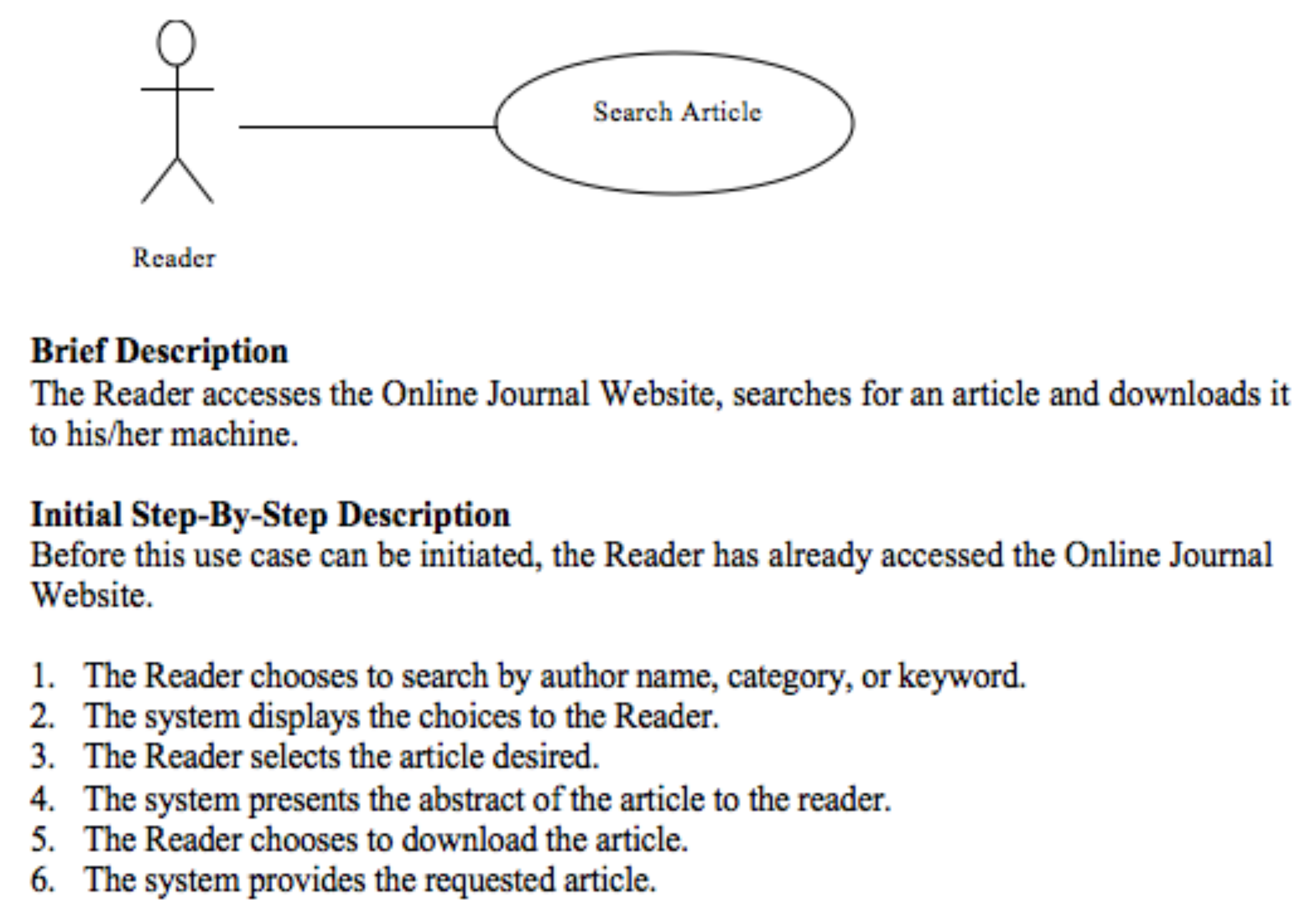
Scope

- Web publishing system for an editor
- Goal: Maximize editor's productivity
  - Automated article review and publishing
- Facilitate communication between the editor, a group of reviewers and the author via email
  - Preformatted reply forms to provide uniform reviewing process

System Environment Diagram



Use Case 01: Search Article

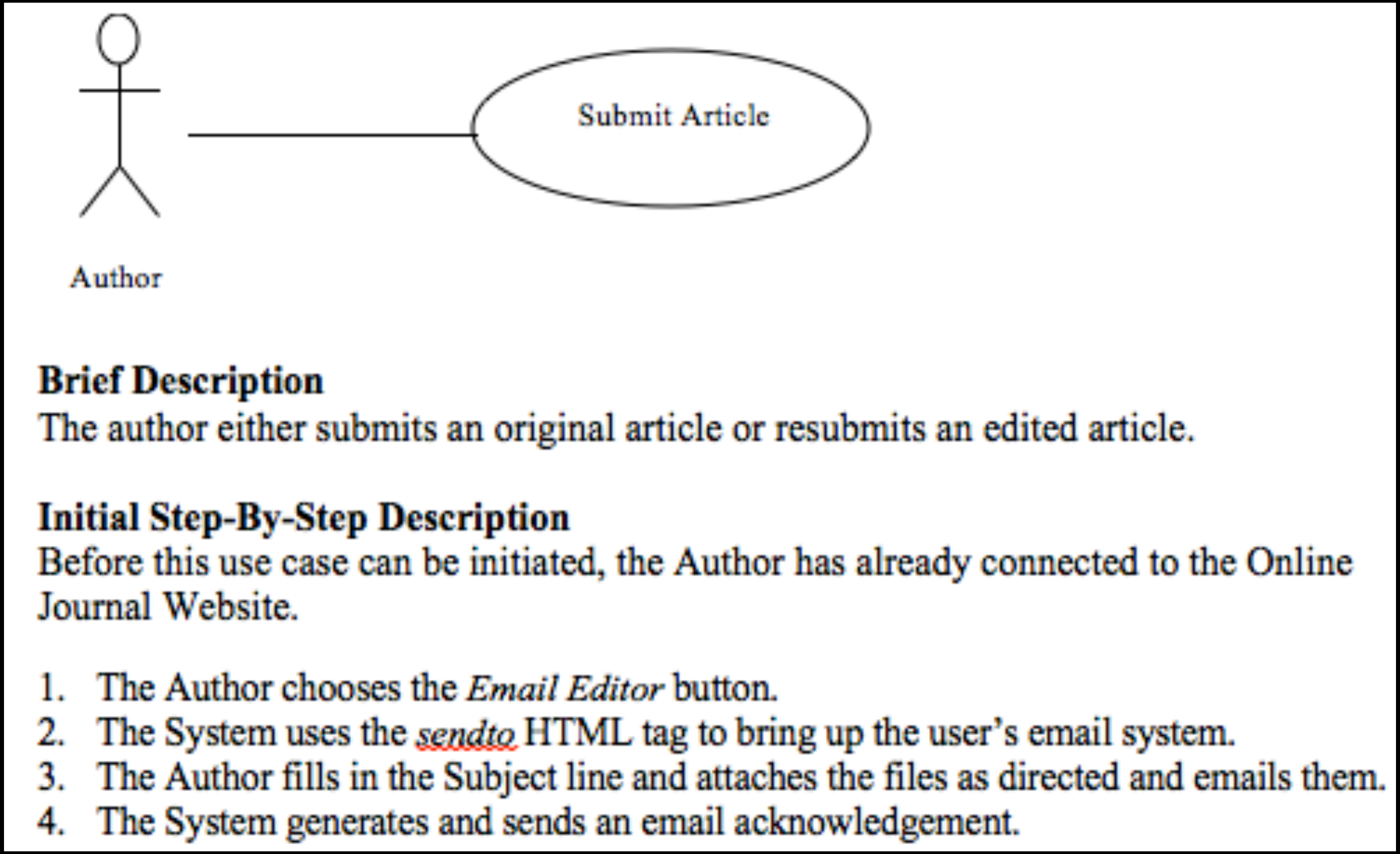

Search Article
Reader

Brief Description
The Reader accesses the Online Journal Website, searches for an article and downloads it to his/her machine.

Initial Step-By-Step Description
Before this use case can be initiated, the Reader has already accessed the Online Journal Website.

1. The Reader chooses to search by author name, category, or keyword.
2. The system displays the choices to the Reader.
3. The Reader selects the article desired.
4. The system presents the abstract of the article to the reader.
5. The Reader chooses to download the article.
6. The system provides the requested article.

Use Case 02: Submit Article

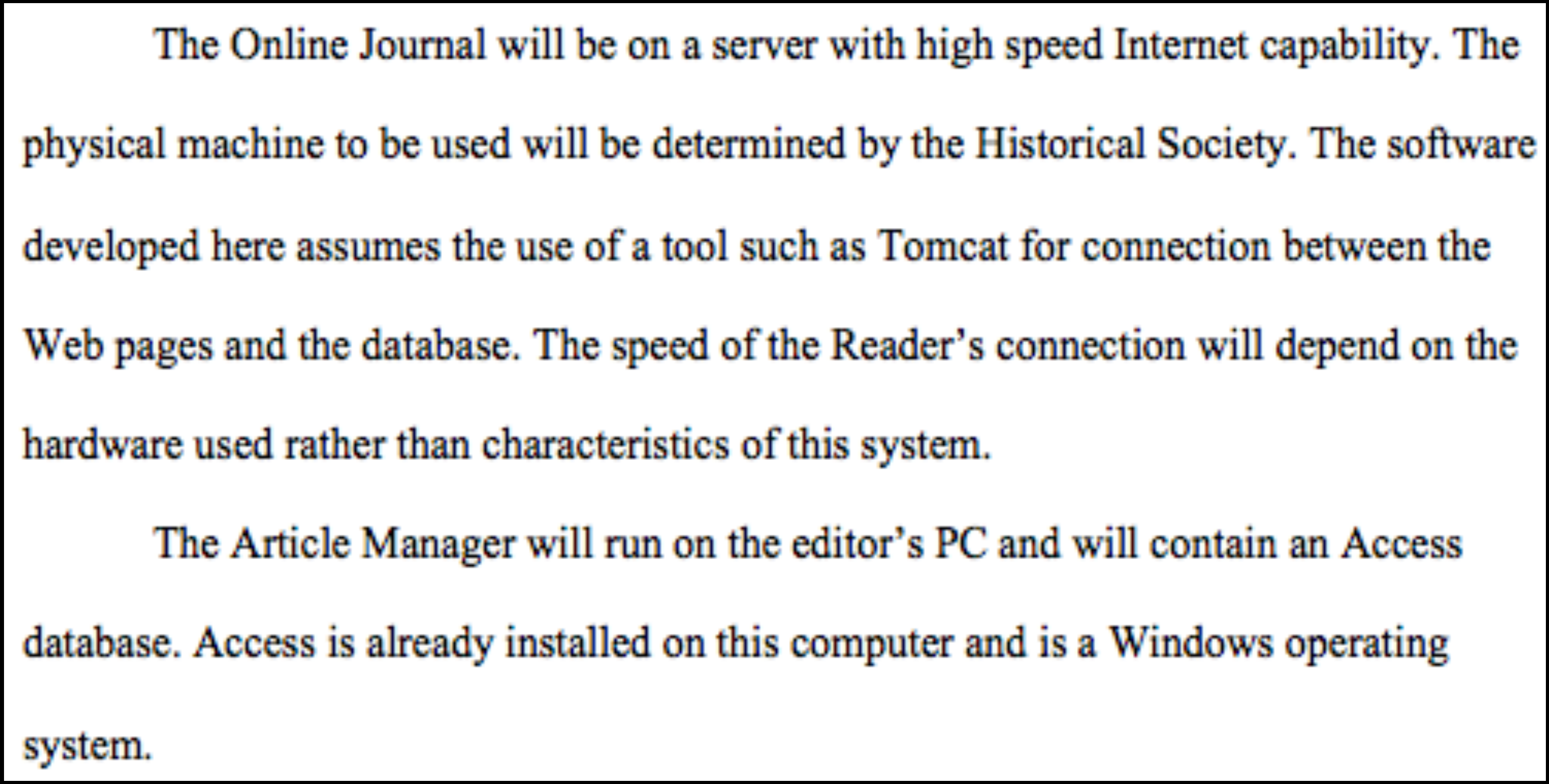

Submit Article
Author

Brief Description
The author either submits an original article or resubmits an edited article.

Initial Step-By-Step Description
Before this use case can be initiated, the Author has already connected to the Online Journal Website.

1. The Author chooses the *Email Editor* button.
2. The System uses the *sendto* HTML tag to bring up the user's email system.
3. The Author fills in the Subject line and attaches the files as directed and emails them.
4. The System generates and sends an email acknowledgement.

Non-functional Requirements



The Online Journal will be on a server with high speed Internet capability. The physical machine to be used will be determined by the Historical Society. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database. The speed of the Reader's connection will depend on the hardware used rather than characteristics of this system.

The Article Manager will run on the editor's PC and will contain an Access database. Access is already installed on this computer and is a Windows operating system.

Your SRS Outline

- Follow the information on the web
  - mostly straightforward, but answer questions thoroughly
- Write in complete sentences and paragraphs
- Explain your choices, describe your product as if we have not seen your presentation
- Parts:
  - Product description, Use cases (also in SRS)
  - UI diagrams, process, customer meeting report

Reminders

- Group Meetings
  - Name your team!
  - Select your Program Manager (PM)
  - Schedule your weekly group meetings and let your TAs know your meeting times
    - We won't be able to meet with each group every week, but we will try our best and coordinate with your PMs
    - Also you can always setup appointments with us
- SRS document due on April 10 (next Tue.) @11PM