TAYLOR'S UNIVERSITY

Wisdom · Integrity · Excellence

SCHOOL OF COMPUTING AND IT

CAPSTONE PROJECT 1 (BACHELOR OF COMPUTER SCIENCE)

JMS for JESTEC

JACKY YAP

(0327868)

School of Computing and IT

MARCH 2020

Supervisor: SUKHMINDER KAUR

THIS REPORT IS DONE IN PARTIAL FULFILLMENT OF THE COURSE REQUIREMENTS FOR THE

Captone Project 1 (Bachelor of Computer Science)

ACKNOWLEDGEMENT

The author would like to express his utmost gratitude to Taylor’s University for providing the author with the opportunity ………………………………………..

Throughout this project………………………………, the author ………………. In addition, ………

.

COPYRIGHT

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work (except Chapter 1 that is common across the project group);
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every right in the copyright to this Work to Taylor's University (TU), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of TU having been first had and obtained;
(6) I have seemed appropriate permission from the Industry Partner (Client) before including any material(s) that may be deemed as proprietary or sensitive in nature.
(7) I am fully aware that in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by TU.

Candidate's Signature   Date:

Subscribed and solemnly declared before,

Witness's Signature   Date:

Name:

Designation:

ABSTRACT

Abstract start here……………………………………………………………….….

…………………………………………………………………………………………………

……………………………………………

TABLE OF CONTENTS

LIST OF TABLES

1. Project Description

1.1 Project Background

Our team is tasked with the development of a Journal Management System (JMS) for Taylors university research journal JESTEC. We aim to implement a web cloud-based solution that will be replacing the current system. Features to be included and commercialized are the submission, review, and acceptance of journal/manuscripts online with automated cloud support. The expected results are to increase the traffic and revenue by speeding up the journal submission/review process. Our goal is to deploy a journal submission system that is completely online using web and cloud services. The process takes place as an author that's trying to submit their research paper, although before doing so, they'll be greeted with the requirements and the format that the paper needs to be in before it goes through the reviewing phase. Even so, the author would be expected to already have made an account on the website, if not, then it'll be instructed to them, the website aims to be more friendly towards the user based on its simplicity and it's a step by step guide on showing you how to submit your research paper into the review process. The guidelines consist of its figures, equations, tables, nomenclature, and the limit of the paper itself. After that is done, the editor will receive a notification from the website notifying the submission of a paper that has taken place, and it will require the attention of the editor within the week. Assuming the editor checks the submission of the paper straight away, the paper will then be assigned to an associate editor, as he's the one responsible for picking out the reviewers, the reviewers will have 4 weeks to review the paper if they choose to accept review process. Reviewers will receive an email for the request to review. After the review is done, it'll be sent back to the associate editor on the comments regarding the paper if there is any, if revision is needed, an email will be sent to the author by the associate editor, regarding the faults that were made in the paper, and needs to be reworked, the author has 4 weeks to work on it when everything is done, the associate editor will have the final decision to accept the paper or to reject, hence the revision. Once the paper has been submitted, the associate editor will send the paperback to the editor with formatting the paper to the current standard of the website, and the editor will request the author to get a copyright for the research paper to be submitted to the website.

JMS will be developed similar to JESTEC's current system, as we want to make sure that current users will easily transition to the new system, but on the backend part of the system it'll be completely different as we've said before, it'll be a cloud-based system that automates most of the work that previously one of the stakeholders has the responsibilities of doing so, things like storing elements of the website, for example, the submitted research papers, will be stored in the cloud. Unlike now, it is stored on one of the stakeholder's machines, which we find not so easy for all the 4 clients, as all the files will be going through him first. Other than that, we also aim to remove the clutter that's existent on the website, if you head to JESTEC, there will be a lot of information served at your face, and we think that it might be demotivating for some people, that's why we want our website to be user friendly, guiding the user step by step on the process of submitting a paper, simplicity will be a big factor in JMS.

1.2 Problem Statement

The current JMS process is handled manually, including all communication. From the moment of manuscript submission, the review process can take well over a month to complete a result. During this, there is communication between multiple parties necessary.

Thus, we can identify the main issues of the manual process to:

- Communication between parties
- Record of submissions and their status
- Lack of automation (Forms, messaging, etc.)
- Tracking of workflow
- Search and filter information (reviewers, submissions, etc.)

Validation and reliability of data could be compromised since processes are not automated, human errors are inevitable. Without the ability to validate automatically an error could lead to delay of the entire process. Especially with the recurring communication between multiple parties.

Automation allows the system to minimize the input needed by the user, thus reducing human error. Additionally, it would allow information to be stored effectively, allowing for validation as well as the reliability of the data collected.

The login system for people involved is completely missing which does not allow for validation of the user, furthermore, processes must be tracked by the individual themselves. Without the knowledge of the workflow, this can prove to be a challenge and prone to additional errors.

Communication is currently handled entirely through email. With deadlines constantly having to be meet it tracking and sending emails to individuals can be challenging. Email addresses of the individuals involved must keep track of manually. Lastly, an issue with the use of personal email is the clutter in the mailbox, which does not allow for specialized categorization for JMS purposes.

The ability to **search and filter** for information would improve the time needed for the user to find or see what they need to see. This would help increase the speed and ease of use of the system. Reducing the visual clutter to what is most important to the user at the time.

Due to the nature of the current system being fully manual, the manual storage of information does not allow for analytics to be run on all the data sets. Analytics can allow the user to gain further insight and reveal information previously unclear.

The current website does not adhere to the web design principles [6]; information is scattered throughout the website, causing the website to look cluttered and unpleasant to look at. This also makes it harder for users to navigate through the website and find the information that they are looking for

1.3 Project Objectives

Table 1.1 Project objectives

No.	Proposed Functionality	Problems Solved/ Opportunities
1	Website redesign	Web design principles[6]
2	User Login/ Sign-up	Login system[3]
3	Author functionality	Automation[2], Search and filter[5]
4	Editor functionality	Automation[2], Search and filter[5]
5	Reviewer functionality	Automation[2], Search and filter[5]
6	System admin functionality	Automation[2]
7	Messaging	Communication[4]
8	Template management	Automation[2]
9	Workflow/ Job counter	Automation[2]
10	Storage of data	Validation and reliability of data[1], Automation[2]
11	Threat Modelling	Security issues
12	Penetration Testing	Security issues
13	Unit Testing	Validate that each unit of the software performs as designed
14	Integration Testing	Validate that individual units of the software performs well when joined as a group
15	System Testing	Validates the complete and fully integrated software product
16	Acceptance Testing	Validate the system's compliance with the business requirements and determine whether it is acceptable for launching

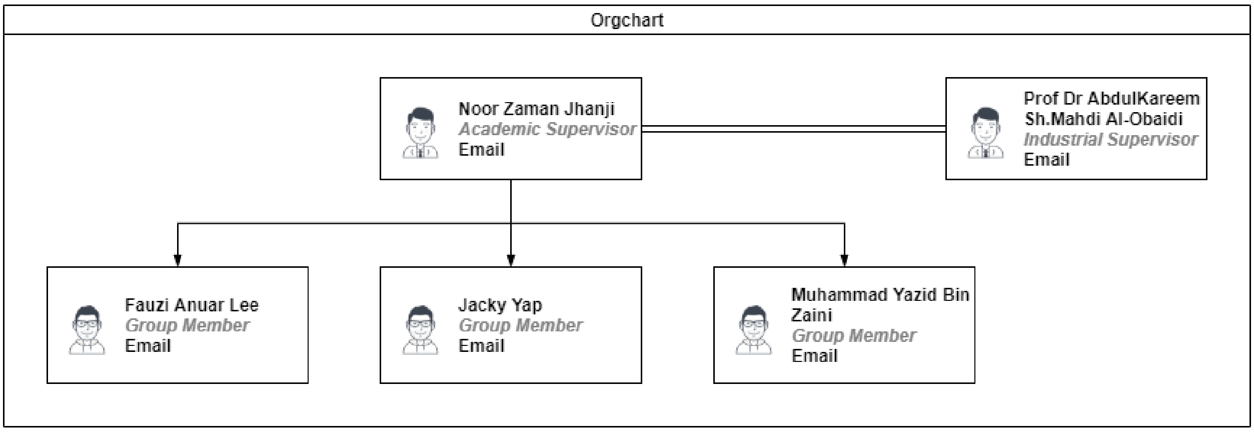
1.4 Stakeholders

Table 1.2 Stakeholders

Stakeholder	Role(s)	Responsibilities
Prof Dr AbdulKareem Sh.Mahdi Al-Obaidi	Project Sponsor	To provide information on the scope, monitoring the progress, and providing any other assistance that is required.
Noor Zaman Jhanjhi	Project Supervisor	To provide guidance and support in term of knowledge, opinion and view throughout the entire project. Furthermore, will be evaluating the project and provide recommendation to continuously improve the project.
Fauzi Anuar Lee	• UI/UX Designer • System Analyst • Programmer • Tester	Lead UI/UX designer. Analyze the visual hierarchy of the website and research UI elements of existing systems. To analyze and research on system requirements. To implement the proposed solutions into functional software. To test and validate the system.
Jacky Yap	• Data Engineer • System Analyst • Programmer • Tester	Design and implement the data base schematics. To analyze and research on system requirements. To implement the proposed solutions into functional software. To test and validate the system.

1.4.1 Organization Chart

Table 1.3 Organization Chart



1.5 Project Scope

The JMS will be developed for use on a web platform with cloud solutions. The 5 main users will be able to access all functions respectively through the website. The 5 primary users are:

1. **Editors**
    a. Receive manuscripts
    b. Assign manuscripts to Associate Editors
2. **Associate Editors**
    a. Assign reviewers
    b. Send reminders
    c. Approve or deny manuscripts
3. **Authors**
    a. Submit manuscripts
    b. Nominate reviewers
    c. Receive review feedback
    d. Submit copyright transfer form
4. **Reviewers**
    a. Review manuscripts
5. **System Admins**
    a. Publish approved manuscripts on website

The last users to be considered are readers that visit the website to browse through the publications.

The creation of this solution aims to provide a platform for journals to be reviewed, published and read. In order to deliver a great experience for all types of users, we will streamline the workflow and automate numerous tasks. For our JMS, we want it to be intuitive to use, without the need for extensive training. In order to achieve this, we reduce visual clutter and provide a clear sequence in workflow and bring attention to tasks that immediate feedback.

Database

To create a database system that suits the needs for this JMS. A database will allow the storage of all information regarding the JMS. It would store data that will be used in the website such as the journals published as well as reviewer's information. A database allows for easy storage of information from automated forms as well as the access to it through queries and other means.

Security system

The latest security system will be applied into the website to prevent malicious parties from accessing sensitive data stored in the database. For instance, to ensure only the authorized users can access and edit the database, each user will be given different levels of access. The website application will also use two-factor authentication (2FA) to further secure the system.

Web platform

Will provide an environment for the JMS and its users. We have created a sitemap to showcase the hierarchical order of the webpage. An added benefit of creating the sitemap is to enable Google search engine crawlers to index our website and its content; this allows our website to appear on their search engine and be found easily.

Landing Page

Redirects user to pages that display information about

- How to submit a paper
- Index & awards
- List of editors
- List of reviewers
- Archives of past issues

Dashboard

The Dashboard is the page where most of the system’s functions are accessed

- Information about the manuscripts involved with the user is displayed here. The information displayed on this page varies from user to user (author, reviewer, editor).
- Among the most important information displayed here to the user is the manuscript status.
- Users can filter their manuscripts according to status.
- By clicking on the desired manuscript, a pop-up will appear where they are able to perform various tasks concerning the manuscript.

Manuscript Pop-up

The functions on this pop-up vary from user to user (author, reviewer, editor). Editors can:

- Contact author(s)
- Assign editor
- Assign/add reviewer
- Change status
- Send message
- Read paper
- Review paper

Whereas reviewers are only able to:

- Send message
- Read paper
- Review paper

Send Message

From the pop-up page, users can send messages to relevant parties involved with the manuscript.

- Users can send various pre-determined messages to other parties (authors, editors, reviewers).
- Information stored in the database will be used to fill in the blanks in these messages.
- With this, users will not have to take the time to draft a message every time.
- Only editors can edit their messages before sending them.

Messages Page

Messages are compiled based on their respective manuscripts. This way, users can easily keep track of the messages they have received about them.

1.6 Risk Management

Table 1.4 Risk Management

Risks	Impact Level	Counter Measurements
Client wants to change design of website	Low	- Make sure that the code foundation of the website is easy to be updated or replaced - Make sure that the client is satisfied with the design before we start developing
Inexperienced team in cloud computing	Medium	-The lack of experience in indulging ourselves in cloud computing can be quite risky for the project, but we believe in the platform we have chosen to make it as easy as possible for us to transition into the cloud computing era.
Time given for the project	Medium	-We are requesting an opportunity from Taylor's to let us intern there while working on the project with the budget required to further improve the quality of the project. -If that does not work out, we plan to start working on it as soon as possible while having applications like Trello and Google Meet for making sure everything is going as planned.
MCO Lockdown Extensions	Low	-If the world is back in shock with numbers rising, we will decide to use Google Meet for frequent meetings on the advancements of the project. -Productivity overall will be lower as we can't have physical meetings and the ability to experiment things in the computer lab will be unavailable.

1.7 Work Breakdown Structure

1. Project Initiation

1.1. Project Background

1.2. Problem Statement/Opportunities

1.3. Project Objectives

1.4. Stakeholders

1.5. Project Scope

1.6. Risk Management

1.7. Work Breakdown Structure

1.8. Activity List

1.9. Responsibility Assignment Matrix

1.10. Project Scheduling

2. Literature Review

2.1. Technology

2.2. Cloud Service Providers

2.3. Domain Research

3. System Analysis

3.1. Current System Analysis

3.2 Dataflow Diagram

3.3. Proposed System

3.4. Use Case Diagram

3.5. Use Case Specification

3.6. SWOT Analysis

3.7. Workflow diagrams of existing system

System Design

4.1. Interface Design

4.2 Database Design

4.3. Workflow of proposed system

4.4. Class diagram

4.6. Design sequence diagram

4.7. Test plan design

4.9. Pseudocode

1.8 Activity List

1. Project Initiation

- ***1.1. Project Background***
- ***1.2. Problem Statement/Opportunities***
- ***1.3. Project Objectives***
- ***1.4. Stakeholders***
- ***1.5. Project Scope***
- ***1.6. Risk Management***
- ***1.7. Work Breakdown Structure***
- ***1.8. Activity List***
- ***1.9. Responsibility Assignment Matrix***
- ***1.10. Project Scheduling***

2. Literature Review

- ***2.1. Technology***
  - *2.1.1. Implementation Language*
    - 2.1.1.1. HTML
    - 2.1.1.2. CSS
    - 2.1.1.3. Ruby
    - 2.1.1.4. MySQL
  - *2.1.2. Tools*
    - 2.1.2.1. Ruby on Rails
    - 2.1.2.2. Bootstrap
    - 2.1.2.3. Microsoft Security Graph
    - 2.1.2.3. MySQL Security
    - 2.1.2.4. GitHub
- ***2.2. Cloud Service Providers***
  - *2.2.1. Microsoft Azure*
  - *2.2.2. Amazon Web Services (AWS)*
  - *2.2.3. Google App Engine*
  - *2.2.3. Price comparison*
- ***2.3. Domain Research***
  - *2.3.1. Analyse peer-review process*

1.9 Responsibility Assignment Matrix

Table 1.5 Responsibility Assignment Matrix

Responsibility	Fauzi	Jacky	Yazid
1. Project Initiation			
1.1. Project Background	✓	✓	✓
1.2. Problem Statement/Opportunities	✓	✓	✓
1.3. Project Objectives	✓	✓	✓
1.4. Stakeholders	✓	✓	✓
1.5. Project Scope	✓	✓	✓
1.6. Risk Management	✓	✓	✓
1.7. Work Breakdown Structure	✓	✓	✓
1.8. Activity List	✓	✓	✓
1.9. Responsibility Assignment Matrix	✓	✓	✓
1.10. Project Scheduling	✓	✓	✓
2. Literature Review			
2.1. Technology			
2.1.1. Implementation Language			
2.1.1.1. HTML	✓		
2.1.1.2. CSS	✓		
2.1.1.3. Ruby	✓		
2.1.1.4. MySQL		✓	✓
2.1.2. Tools			
2.1.2.1. Ruby on Rails	✓		
2.1.2.2. Bootstrap	✓		
2.1.2.3. Microsoft Security Graph			✓
2.1.2.3. MySQL Security			✓
2.1.2.4. GitHub	✓		
2.2. Cloud Service Providers			
2.2.1. Microsoft Azure		✓	✓
2.2.2. Amazon Web Services (AWS)		✓	✓
2.2.3. Google App Engine		✓	
2.2.3. Price comparison		✓	
2.3. Domain Research			
2.3.1. Analyze peer-review process		✓	
2.3.1. Analyze previous security model			✓
2.3.1. Identify security solutions			✓
2.3.2. Case Studies		✓	✓
2.3.3. Web design principles & usability	✓		

3. System Analysis			
3.1. Current System Analysis		✓	
3.2 Dataflow Diagram			✓
3.3. Proposed System	✓		
3.4. Use Case Diagram	✓	✓	
3.5. Use Case Specification	✓	✓	
3.6. SWOT Analysis			✓
3.7. Workflow diagrams of existing system			✓
4. System Design			
4.1. Interface Design			
4.1.1. Archive Page	✓		
4.1.2. Journal Article Reader Page	✓		
4.1.3. Dashboards	✓		
4.1.4. Manuscript Reader & Review Form	✓		
4.1.5. Messages Feature	✓		
4.2 Database Design			
4.2.1. Entity Relationship Diagram (ERD)		✓	
4.2.2. Data Dictionary		✓	
4.2.3. MySQL Implementation		✓	
4.3. Workflow of proposed system			✓
4.4. Class diagram			✓
4.6. Design sequence diagram			✓
4.7. Test plan design	✓		
4.9. Pseudocode			✓

1.10 Project Scheduling

Table 1.6 Project Scheduling

No.	Activities	Days	Start Date	Finish Data
1.	Requirement Gathering	7	Week 1	Week 2
2.	List objectives	7	Week 2	Week 3
3.	Survey old website	14	Week 1	Week 3
4.	Client comment specification	14	Week 1	Week 3
5.	Threat Modelling	7	Week 3	Week 4
6.	Client review specification	60	Week 1	Week 9
7.	Prototyping	60	Week 1	Week 9
8.	Domain Research	7	Week 3	Week 4
9.	Implementation Language Research	7	Week 3	Week 4
10.	Research on Tools needed	7	Week 3	Week 4
11.	Cloud service provider needed	7	Week 3	Week 4
12.	Design the dataflow of the project	7	Week 4	Week 5
13.	Create proposed system	7	Week 4	Week 5
14.	Specify use case scenario	7	Week 4	Week 5
15.	Analyse the SWOT	7	Week 5	Week 6
16.	UI/UX designing	60	Week 4	Week 12
17.	Database designing	14	Week 4	Week 6
18.	Creating the ERD	14	Week 5	Week 7
19.	Creating the Design sequence Diagram	7	Week 3	Week 4
20.	Analysing the workflow of proposed system	7	Week 3	Week 4
21.	Creating class diagram	7	Week 10	Week 11
22.	Creating the test plan design	7	Week 7	Week 8
23.	Listing out the pseudocode	7	Week 8	Week 9

2. Literature Review

2.1 Technology

2.1.1 Implementation language

2.1.1.1 Introduction

The traditional process of peer review for scientific papers has evolved into a digital form with the information age. The demand for a platform that facilitates this process online is increasing (Lederman & Lederman, 2017). This advance in technology enables researchers to have their papers published with less cost involved.

We want to create a Journal Management System (JMS) that automates the peer review process using a web-cloud based solution. As part of our capstone project, we conduct this research on the procedures and technologies involved to gain a deeper understanding. From the previous requirement gathering process we have identified problems that require further research for us to effectively resolve them.

We concluded that communication between editors and reviewers in the peer review process is the main problem to be solved. To maximize productivity, instant communication is key to shorten the time needed to complete the review process. A platform that facilitates communication and provides automation will be required.

The other issue is the validation and reliability of the data, an error could lead to a delay or halt of the entire process. Finding the right methodologies for storing and retrieving our data is needed. Automation and validation will result in more consistent data, as traditional ways are prone to human error.

We will start with research into existing Open Journal Systems (OJS) which could lead us to additional insights to discover problems that we have not yet considered. Since our previous research is based solely on the requirements of the system given by our industrial advisor.

Followed by a comparison of SQL and NoSQL database technologies to gauge which is more suitable for our purposes. We assume that the driving factor for success will be the speed and reliability we can store and retrieve data.

This literature review will further advance our project research by helping us discover more about OJS. In a time where technology and knowledge advance at a rapid rate we must utilize everything we can to improve the process that verifies our knowledge.

2.2 Domain research

2.2.1 Analyze peer review process

As a take into the future of the peer review process we look from the perspective of the users the peer review process relies on. The review process requires multiple individuals; Editors and reviewers are expected to execute their tasks professionally and on time. The work done is often for non-profit and most of the time will come with no compensation. Lederman stated, “Being a reviewer is a thankless job, with the only tangible reward being that you can list your editing experience as an item on your vita” (Lederman & Lederman, 2017, p. 3). As a result, the base of reviewers might not shrink, but the quality of reviews could drop. Reviewers might decline invitations more or they are late with their review deadlines (Lederman and Lederman 2017, pp. 3-4). Knowing that reviewers are essential in a peer review process we must address these issues and develop a solution to retain a large group of reviewers and stop the reviewer base from shrinking. For us to conduct a fair evaluation we require a variety of different viewpoints and angles (Lederman and Lederman 2017, p. 4).

Additionally, (Tracz and Lawrence 2016, p. 2) state that “There is some recognition amongst the research community that journals are now an outdated method for publishing new research findings and no longer serve the needs of much science”. One of the problems given is that the peer review process being placed before the publication of journals and thus, the progression of science (Tracz and Lawrence 2016, p. 2). Another is the non-transparency of the review process between author and reviewer (Tracz and Lawrence 2016, p. 2). Often review is anonymous to the author, with the details of the review itself extruded. This also leads to the readers being deprived of the information of the reviewer (Wicherts, 2016). In research where publications can improve one's career, this leaves room for abuse of the system (Tracz and Lawrence 2016, p. 2).

2.2.2 Case Study

An experimental model is proposed (Tracz & Lawrence, 2016) that allows for publications to be made as soon as possible after submission. Peer review is only conducted after publication with a list of formally invited reviewers, these types of publications are known as preprints. Reviewers will conduct the process transparently and share with the author how their article can be improved (Tracz & Lawrence, 2016). Allowing the reviewer to earn credit for their contribution in the form of a digital object identifier (DOI) that allows them to be cited. This will increase the willingness of people to provide their work (Tracz & Lawrence, 2016). A further benefit of credentials earned; it can add to the assessment instead of being lost to the public (Tracz & Lawrence, 2016). Qualitative and quantitative measures that fit the purpose of the research should be used to determine the quality, instead of set metrics. All indicators should be considered to set the value of a journal. This ensures that all knowledge is utilized immediately with no delay (Tracz & Lawrence, 2016). An example of a platform that utilizes preprints is F100. They provide tools that aid in the creation and assessment of qualitative research. Although other publishers have adopted preprints, many still rely on the traditional way of publishing journals. For true open peer review to be possible there must be a major shift in the way research findings are reviewed (Tracz & Lawrence, 2016).

Next, we take a look at the Korean Journal Publishing Service (KPupS), which is using full-cycle open access (OA) publication model and platform prototype for distributing research outcomes in science, technology, engineering, and medicine (STEM) in South Korea (Park & Seo, 2016). This involves a total of 4 phases, those are creation, digital archiving, web service, and circulation. The phases were designed with consideration of case studies from various institutions. Those include Bepress a publisher of academic journals from the University of California (Park & Seo, 2016). As well as Open Journal Systems (OJS), which is an open-source program that manages the complete reviewing and publication process (Park & Seo, 2016). The case studies concluded that a wide range of different digital journal publishing services is moving towards the OA model (Park & Seo, 2016).

There are a variety of different types of OA models. The two we will discuss are Green Open Access and Gold Open Access. The main differences are for the green access authors will assign copyright to the publisher, while retaining the right to disseminate an OA copy of the accepted manuscript to OA repositories (Gargouri et al., 2012). Gold access authors will instead retain their copyright while releasing a license to publish to the publisher, but Gold Open Access may require the payment of an article processing fee to the publisher. This comes with the benefit of articles being released to the public immediately. Green Open Access might embargo accepted papers for a time. Giving access only to subscribers of the publishing service. Later, after the embargo period papers can be downloaded from OA repositories (Gargouri et al., 2012). Another insight from the case studies showed that with the increase of publication through electronic services, the representative electronic service is DOI, which assigns a unique identifier to the article. This increases the reference linking on the internet. Other technologies also allow for metadata to be provided to academic journal systems. This provides them with a wider availability through various search engines (Park & Seo, 2016).

Overall, the peer review process relies on the contribution of its editors and reviewers to ensure the reliability and quality of the journals. Authors are important contributors that provide content to the publisher. Thus, it is necessary to improve and ease the work process of said entities. By crediting the work reviewers do, we give them incentives to provide the publisher with timely and quality reviews. Similarly, a digital object identifier (DOI) for the articles published can achieve additional incentive for the author. Furthermore, having an effective user interface eases the process of everyone involved through guided processes, templates, and standardized forms. Having a platform and digital object identifier (DOI) will also allow for metadata to be collected; this can help increase the availability of the Journal publisher, the site and its content will be easier to find through various search engines.

2.3 Technology

2.3.1 Comparative study SQL and NoSQL

For us to store the data of our JMS, we must choose the type of database we want to use. We will be looking at SQL and NoSQL databases. Each has its strengths and weaknesses. Starting with SQL or relational database, which is commonly used in traditional database management systems (Sahatqija et al., 2018). Data is stored away in tables and isolated in lines consisting of records and fields that contain the information. The schemas are predefined which allows for data integrity and removes data redundancy, assuming all tables have been fully normalized (Sahatqija et al., 2018). Using their standard query language information can be retrieved. This makes the relational database management system simple to use overall (Sahatqija et al., 2018). Another main feature of SQL database are the ACID properties which stands for Atomicity, Consistency, Isolation, Durability (Ali et al., 2019). When it comes to transactions that need high reliability such as banking, finance, or security systems ACID is essential (Ali et al., 2019). Depending on the requirements RDBMS might not be suitable for; Distributed databases over a wide network or when low latency is required (Ali et al., 2019). Other requirements that do not fit RDBMS include Scalability, Control over performance characteristics, high availability, low latency, and is cheap to implement (Ali et al., 2019).

These weaknesses gave birth to NOSQL which stands for Not Only SQL. This type of database provides means for storing and retrieving data in different from the relational approach of storing data in tables. NoSQL or non-relational databases are intended to be able to accommodate a large rate of data. Being able to scale horizontally using additional documents is a big advantage. There are several types of NoSQL databases: Document, graph, key-value, or wide-column based. When it comes to big data analytics NoSQL databases can handle the general features of big data which are:

- High data velocity, where information is streamed from distinct locations and sources at a rapid rate and with continuous updates.
- Data Variety, meaning that the type of data stored could be structured, unstructured, or semi-structured.
- A high volume of data is generated and stored. The data sets can range from terabytes to petabytes.
- The complexity of information is high because the data is stored at various sites or information centers.

Considering these features NoSQL is highly suitable for the big data environment.

Both SQL and NoSQL have their advantages and drawbacks over each other. In general, we can conclude that SQL is highly suited for transactions where ACID properties are essential. NoSQL on the other hand can store and retrieve a vast amount of data, which makes it suitable for big data applications. When it comes to deciding which type to use, we must consider and thoroughly understand the requirements of the system we want to implement. This will help us decide which system is best suited for our purposes. To conclude we will summarize the benefits and drawbacks of each in points:

NoSQL Benefits

- Allows the user to choose from different types of data models
- Easily scalable by expanding horizontally
- Administrators are not necessary
- Some NoSQL DB supplier can provide to manage hardware failure
- Faster, more efficient, and flexible
- Has developed rapidly
- Used for big data applications

NoSQL Drawbacks

- Immature
- No standard query language
- ACID compatibility is not possible in some NoSQL Databases
- No interface specification
- It is hard to maintain
- Less support provided

SQL Benefits

- It is simple to use
- Easy to design execute, maintain and use
- One of the main advantages of SQL is that information is only saved in one place
- It offers a variety of interfaces
- It improves the integrity of the information
- It is secured in nature
- It has a standard query language

SQL Drawbacks

- Software is costly
- Hardware overheads
- Limitations in Structure
- The lost data can hardly be recovered.
- High availability issue
- Does not support Big Data Applications
- Certain applications are slow in processing

After thorough research, I assume that SQL will be our choice of database. The reason for this choice is that in comparison SQL is easier to use and implement and has a standard query language. Since my team and I have already had experience in implementing a database using SQL, we are confident that we can deliver quality work. Another reason is the integrity of the information that SQL inherently provides. For our system, we will have predefined schemas and queries that will be run without being the object of frequent change. Therefore, we believe that a relational database will be best suited for our system.

2.4 Cloud Service Providers

2.4.1 Price comparison

In the previous section, we have discussed the type of database we would be using, SQL. Although it is highly suited for the requirements of our project, it does come with its own set of drawbacks. This leads us to the next topic to be discussed, Cloud service providers (CSP). This is a model where computing resources are offered over the internet as a scalable on-demand web service (Mukherjee, 2019). This resolves the need for organizations to build their infrastructure, as it would require a large amount of investment (K et al., 2019). This would not be a problem for large enterprises, but for small to medium-sized businesses that would not be feasible (Dutta & Dutta, 2019). Using the services of CSPs we can address the shortcomings of SQL such as the cost of hardware and scaling (Madhuri & Sowjanya, 2007). We will be comparing 3 leading CSPs and decide which will be most suitable for the implementation of our project for JESTEC. Factors to be considered are cost, compatibility, features, and support.

Table 2.1 On-demand Pricing

Instance type	AWS	Azure	Google	AWS Pricing (per hour)	Azure Pricing (per hour)	Google pricing (per hour)
General purpose	m5.xlarge	B4MS	n1-standard-4	$0.192	$0.166	$0.214
Compute optimized	c5.xlarge	F4s v2	n1-highcpu-4	$0.170	$0.169-0.17	$0.1626
Memory optimized	r5.xlarge	E4 v3	n1-highmem-4	$0.252	$0.252	$0.2696

Table 2.2 1 year commitment Pricing with no upfront cost

Instance type	AWS	Azure	Google	AWS Pricing (per hour)	Azure Pricing (per hour)	Google pricing (per hour)
General purpose	m5.xlarge	B4MS	n1-standard-4	$0.123	$0.097	$0.128
Compute optimized	c5.xlarge	F4s v2	n1-highcpu-4	$0.107	$0.099	$0.095
Memory optimized	r5.xlarge	E4 v3	n1-highmem-4	$0.159	$0.156	$0.159

This is a comparative study on the pricing of cloud service providers (CSP). The 3 we will be looking at are Amazon Web Services (AWS), Microsoft Azure, and Google App Engine. This comparison is conducted on the difference in pricing given in 3 different settings. The settings are:

- General-purpose
- Compute-optimized
- Memory-optimized

General-purpose settings specify that computing power and memory are set to standard performance. Compute-optimized prioritizes high computing power while memory-optimized prioritizes high memory settings.

Another comparison is made for the prices with a discount. This is provided by the CSP when a commitment to a subscription is made; the usual minimum commitment to be eligible for a discount is one year, with no upfront payment.

To conclude the tables, we have indicated the highest price in red and the lowest price in green for the given setting. We will start with on-demand pricing:

- For general-purpose, Azure has the lowest price while Google has the highest price
- For compute-optimized Google has the lowest price while AWS and Azure have the same price.
- For memory-optimized Azure and AWS are priced similarly while Google has a higher price.

For discounted pricing with a 1-year commitment and no upfront cost:

- For general-purpose, Azure has the lowest price while Google has the highest
- For compute-optimized Google has the lowest price while AWS has the highest price
- For memory-optimized Azure has the lowest price while AWS and Google have the same price

3. System Analysis

3.1 Current System Analysis

The system that is currently in use by JESTEC facilitates the review process for the papers that have been submitted to the journal. This is done through a website, where authors can submit their papers through email. The stakeholders of this organization are:

- Editors
- Associate editors
- Authors
- Reviewers
- Admin

We will start with discuss the roles of each stakeholder in the sequence of the review process. The review process commences when an author submits a paper for review to the journal. An editor will receive it and check if it is formatted according to JESTEC standards. If there are no issues the editor will then assign an associate editor. The associate editor will then assign reviewers to assess the paper. Reviewers can accept or decline a review assignment. If they accept, they will be able to see the paper and then provide feedback in the form of revision notes back to the associate editor in due time. The associate editor will then decide based on the feedback from the reviewers. The possible options for the associate editor to decide are:

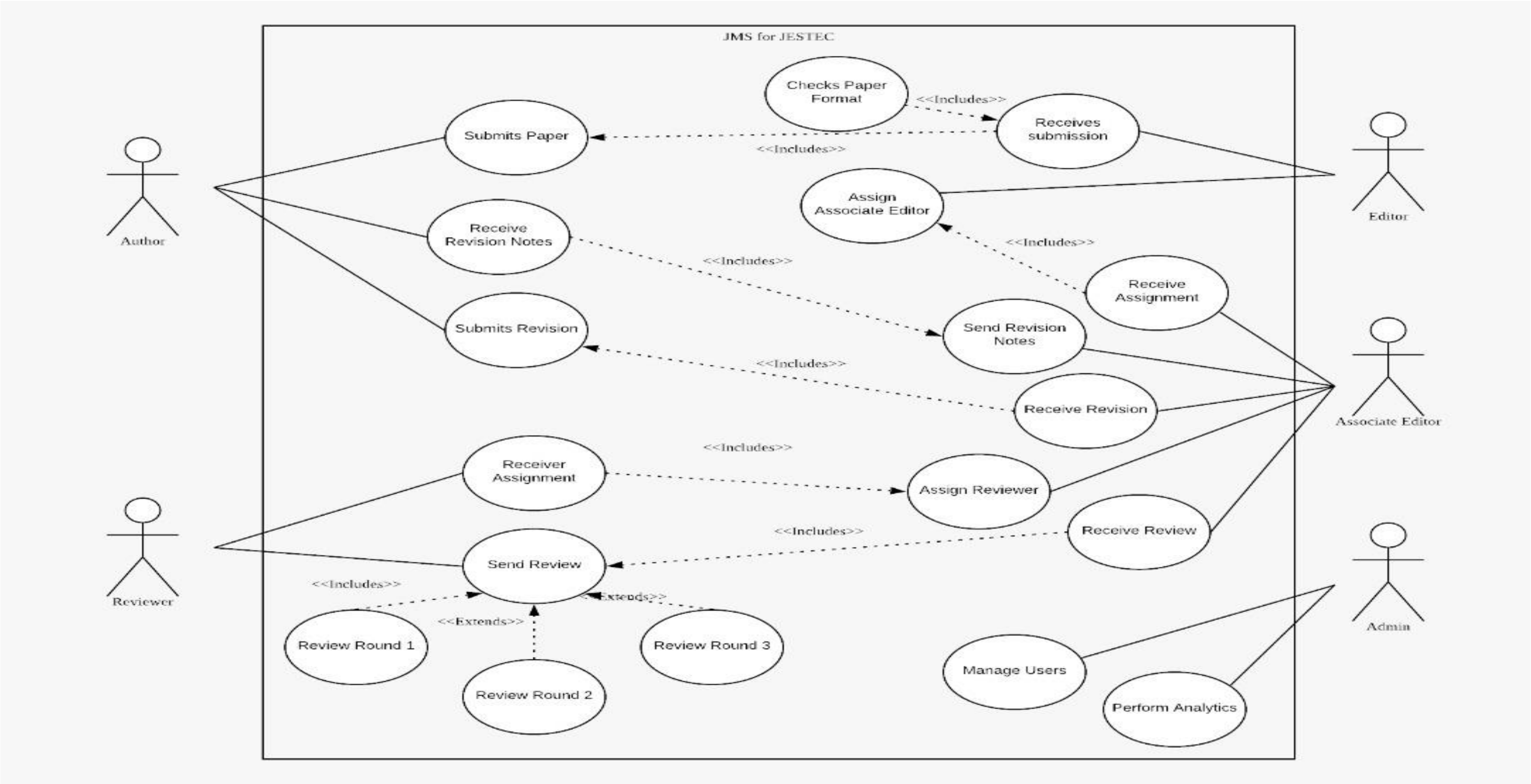
- Accept as is
- Accept with revision
    - Minor revision
    - Major revision

If the paper needs revision, the author will receive the notes from the reviewer and then make changes and resubmit the paper in due time. This revised paper will then go through another round of review with the limit being 3 rounds of review. Then it will either be accepted or declined. The associate editor will then declare it as accepted and prepare the paper for publication. Once published it will be available for view on the website under the archive section.

The technology used for the review system is based on the available website. All communication is handled manually through email. This trend of the manual process also extends to the storing and publishing of the files and data. All of it is saved on the personal machine of senior staff. Considering this process, we can assume the system to be completely manual and only conventional technology such as email and website is used.

3.2 Use Case Diagram

Table 3.1 Use Case Diagram



3.3 Use case Specifications

Table 3.2 Use case specification - Receive Revision Notes

Use Case Name	Receive Revision Notes
Actor(s):	Author (primary), Associate Editor (secondary)
Summary Description:	Allows the author to receive revision notes to make corrections
Pre-Condition:	Associate Editor must send the revision notes to the author
Post-Condition(s):	Author receives the revision notes
Basic Path:	1. Associate editor sends revision notes 2. Author receives revision notes
Alternative Paths:	N/A

Table 3.3 Use case specification - Submits Revision

Use Case Name	Submits Revision
Actor(s):	Author (primary), Associate Editor (secondary)
Summary Description:	Allows the author to submit the revised paper
Pre-Condition:	Author must complete the revision according to revision notes provided
Post-Condition(s):	Associate editor receives the revised paper
Basic Path:	1. The author submits revision 2. The associate editor receives the revision from the author
Alternative Paths:	Revision file submitted by the author has the wrong format

Table 3.4 Use case specification - Receive Assignment

Use Case Name	Receive Assignment
Actor(s):	Reviewer (primary), Associate Editor
Summary Description:	Allows the reviewer to accept assignment for review
Pre-Condition:	Associate editor sent out an invitation to review paper
Post-Condition(s):	Reviewer receives the paper after acceptance of the invitation
Basic Path:	1. Associate editor sends invitation to Reviewer 2. Reviewer accepts the invitation
Alternative Paths:	Reviewer declines the invitation sent by the associate editor

Table 3.5 Use case specification - Send Review

Use Case Name	Send Review
Actor(s):	Reviewer (primary), Associate Editor (secondary)
Summary Description:	Allows the reviewer to send completed review to the associate editor
Pre-Condition:	The reviewer received a review assignment
Post-Condition(s):	The associate editor receives the review from the reviewer
Basic Path:	1. The reviewer completes a review assignment 2. The reviewer uploads the revision notes
Alternative Paths:	1. The reviewer does not send in the review 2. The reviewer uploads the wrong file format

Table 3.6 Use case specification - Review Round 1

Use Case Name	Review Round 1
Actor(s):	Reviewer (primary)
Summary Description:	Allows the reviewer to perform first round of review
Pre-Condition:	The reviewer received an invitation from the associate editor
Post-Condition(s):	The reviewer reviews the paper for the first time
Basic Path:	1. The reviewer receives the information for the paper 2. Assesses the paper and creates feedback in form of revision notes, within 4 weeks
Alternative Paths:	1. The reviewer does not review the paper 2. The reviewer does not review the paper in time

Table 3.7 Use case specification - Review Round 2

Use Case Name	Review Round 2
Actor(s):	Reviewer (primary)
Summary Description:	Allows the reviewer to perform second round of review
Pre-Condition:	The reviewer received the revised paper after the first round of review
Post-Condition(s):	The reviewer reviews the paper for the second time
Basic Path:	1. The reviewer receives the revised paper 2. Assesses the paper and creates feedback in form of revision notes, within 4 weeks
Alternative Paths:	1. The reviewer does not review the paper 2. The reviewer does not review the paper in time

Table 3.8 Use case specification - Review Round 3

Use Case Name	Review round 3
Actor(s):	Admin (primary)
Summary Description:	Allows the reviewer to perform third round of review
Pre-Condition:	The reviewer received the revised paper after the second round of review
Post-Condition(s):	The reviewer reviews the paper for a third time
Basic Path:	1. The reviewer receives the revised paper 2. Assesses the paper and creates feedback in form of revision notes, within 4 weeks
Alternative Paths:	1. The reviewer does not review the paper 2. The reviewer does not review the paper in time

Table 3.9 Use case specification - Manage Users

Use Case Name	Manage Users
Actor(s):	Admin (primary)
Summary Description:	Allows the admin to manage users
Pre-Condition:	The user is an admin
Post-Condition(s):	The admin can make changes to user information
Basic Path:	1. The user logs in as admin 2. The user edits user information
Alternative Paths:	1. The admin does not make changes to user information

Table 3.10 Use case specification - Perform Analytics

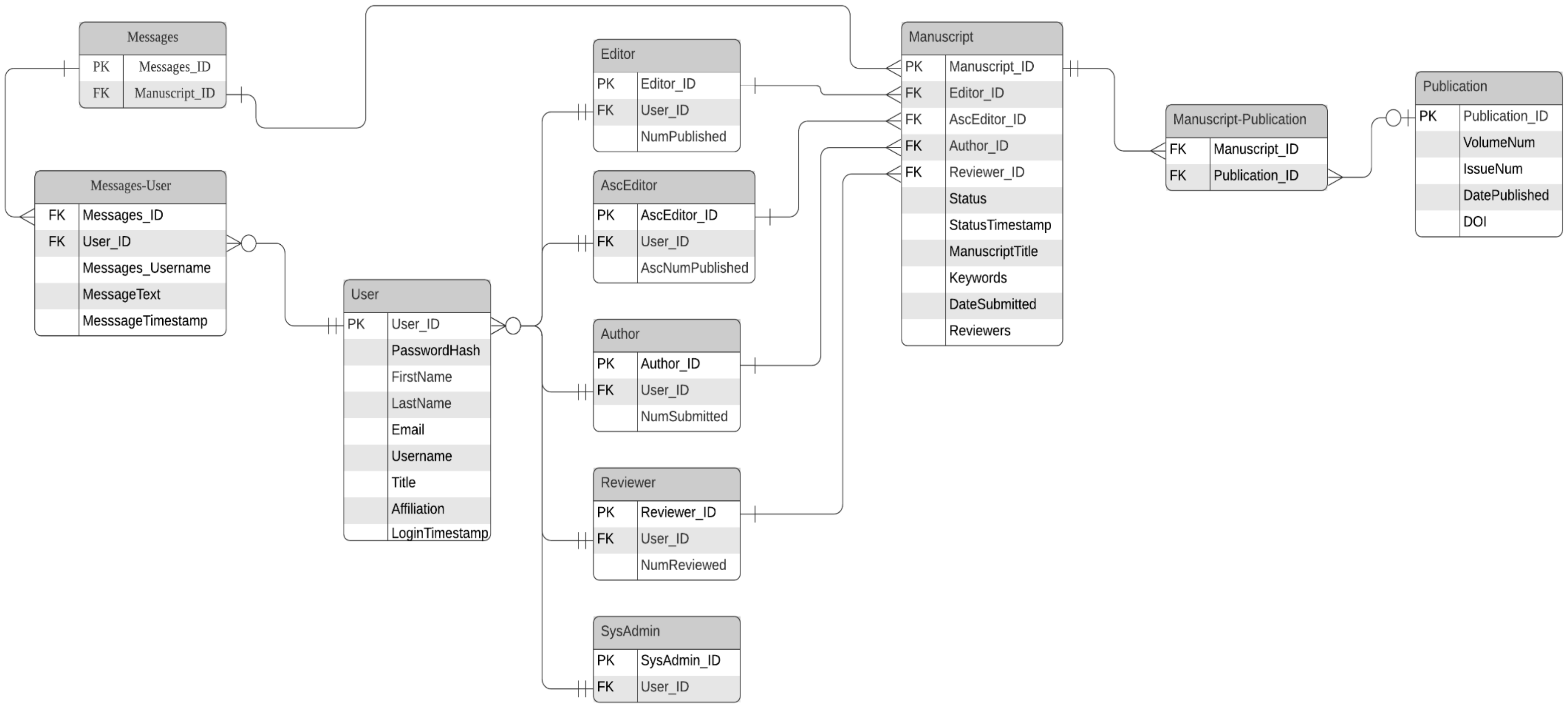
Use Case Name	Perform Analytics
Actor(s):	System Admin (primary)
Summary Description:	Allows the admin to perform analytics on data collected
Pre-Condition:	The user is an admin
Post-Condition(s):	The admin can perform user analytics
Basic Path:	1. The admin logs in as admin 2. The admin creates query 3. Views report
Alternative Paths:	1. The admin does not perform analytics

4. System Design

4.1 Database Design

4.1.1 *Entity Relationship Diagram*

Table 4.1 Entitity Relationship Diagram



4.1.2 Data Dictionary

Table 4.2 Data Dictionary - User

Field Name	Data Type	Data Format	Field Size	Description	Example
User_ID	Integer	NNNNNN	6	Unique ID for all users	000001
PasswordHash	Text	CCCCCCCC	8 to 16	Encrypted password	Ase32Eert
FirstName	Text	LLLLL…	20	First name of the user	Mark
LastName	Text	LLLLL…	20	Last name of the user	Mustermann
Email	Text	CCCCC…	20	Email-address of the user	Mar.M@gmail.com
Username	Text	CCCCC…	16	Preferred handle of the user	Markymark19
Title	Text	LLLLL…	50	Title of the person (Dr., Prof. etc.)	Professor
Affiliation	Text	LLLLL…	50	Institution/organization the user belongs to	Taylors University
UserTimestamp	Miscellaneous	JSON		Date and time when user logs in	1994-12-12 12:34:15

Table 4.3 Data Dictionary - Editor

Field Name	Data Type	Data Format	Field Size	Description	Example
Editor_ID	Integer	NNNNNN	6	Unique ID for users that are editors	000010
User_ID	Integer	CCCCCCCC	8 to 16	Unique ID for all users	123020
NumPublished	Integer	LLLLL…	-2^31-1 to -2^31-1	Total number of papers published by editor	100

Table 4.4 Data Dictionary - AscEditor

Field Name	Data Type	Data Format	Field Size	Description	Example
AscEditor_ID	Integer	NNNNNN	6	Unique ID for users that are editors	000010
User_ID	Integer	CCCCCCCC	8 to 16	Unique ID for all users	123020
AscNumPublished	Integer	LLLLL…	-2^31-1 to -2^31-1	Total num of paper published by AscEditor	100

Table 4.5 Data Dictionary - Author

Field Name	Data Type	Data Format	Field Size	Description	Example
Author_ID	Integer	NNNNNN	6	Unique ID for users that are editors	000010
User_ID	Integer	CCCCCCCC	8 to 16	Unique ID for all users	123020
NumSubmitted	Integer	LLLLL…	-2^31-1 to -2^31-1	Total number of successful submissions by author	100

Table 4.6 Data Dictionary - Reviewer

Field Name	Data Type	Data Format	Field Size	Description	Example
Reviewer_ID	Integer	NNNNNN	6	Unique ID for users that are editors	000010
User_ID	Integer	CCCCCCCC	8 to 16	Unique ID for all users	123020
NumReviewed	Integer	LLLLL…	-2^31-1 to -2^31-1	Total number of reviews by reviewer	100

Table 4.7 Data Dictionary - SysAdmin

Field Name	Data Type	Data Format	Field Size	Description	Example
SysAdmin_ID	Integer	NNNNNN	6	Unique ID for users that are editors	000010
User_ID	Integer	CCCCCCCC	8 to 16	Unique ID for all users	123020

Table 4.8 Data Dictionary - Manuscript

Field Name	Data Type	Data Format	Field Size	Description	Example
Manuscript_ID	Integer	NNNNNN	6	Unique ID for papers submitted by authors	000010
Editor_ID	Integer	NNNNNN	6	Record for Editor involved	000010
AscEditor_ID	Integer	NNNNNN	6	Record for AscEditor involved	000010
Author_ID	Integer	NNNNNN	6	Record for Author involved	000010
Reviewer_ID	Integer	NNNNNN	6	Record for Reviewer involved	000010
Status	Miscellaneous	JSON		Record for current status with timestamp	Submitted ('1970-01-01 00:00:00' UTC) :
ManuscriptTitle	Text	LLLLL…	50	Title of paper submitted by author	Case Study on division by zero
Keywords	Miscellaneous	JSON		Keywords of paper submitted by author	Mathematics; Engineering: …
DateSubmitted	Date/Time	YYYY-MM-DD HH:MM:SS	18	Date and time when paper is submitted by author	1994-12-12 12:34:15
Reviewers	Miscellaneous	JSON		Collection of Reviewer_ID involved in review process of paper	000010: 000012; …

Table 4.9 Data Dictionary - Manuscript-Publication

Field Name	Data Type	Data Format	Field Size	Description	Example
Manuscript_ID	Integer	NNNNNN	6	Unique ID for papers submitted by authors	000010
Publication_ID	Integer	NNNNNN	6	Unique ID for accepted papers ready for publishing	000014

Table 4.10 Data Dictionary - Publication

Field Name	Data Type	Data Format	Field Size	Description	Example
Publication_ID	Integer	NNNNNN	6	Unique ID for papers submitted by authors	000010
VolumeNum	Integer	NNNNNN	6	Volume number assigned to article for publication	000014
IssueNum	Integer	NNNNNN	6	Issue number assigned to article for publication	000010
DatePublished	Miscellaneous	JSON		Date and time of article publication	1994-12-12 12:34:15
DOI	Text		60	Volume number assigned to article for publication	10.1002/0002-8231(199601)47:12.0.TX;2-2

Table 4.11 Data Dictionary - Messages

Field Name	Data Type	Data Format	Field Size	Description	Example
Messages_ID	Integer	NNNNNN	6	Unique ID to identify thread of messages	000010
Manuscript_ID	Integer	NNNNNN	6	Unique ID for papers submitted by authors	000010

Table 4.12 Data Dictionary - Messages-User

Field Name	Data Type	Data Format	Field Size	Description	Example
Messages_ID	Integer	NNNNNN	6	Unique ID to identify thread of messages	000010
User_ID	Integer	NNNNNN	6	Unique ID of the user that send the message	000014
Messages_Username	Text	NNNNNN	6	Username associated with message send	MarkMuster32
MessageText	Miscellaneous	JSON		Content of message	“Reminder of review submission”
MessageTimestamp	Miscellaneous	JSON		Timestamp of message sent	10.1002/0002-8231(199601)47:12.0.TX;2-2

4.1.3 MySQL Implementation

```
CREATE TABLE `Manuscript-Publication` (
  `Manuscript_ID` <type>,
  `Publication_ID` <type>,
  KEY `FK` (`Manuscript_ID`, `Publication_ID`)
);

CREATE TABLE `User` (
  `User_ID` <type>,
  `PasswordHash` <type>,
  `FirstName` <type>,
  `LastName` <type>,
  `Email` <type>,
  `Username` <type>,
  `Title` <type>,
  `Affiliation` <type>,
  `LoginTimestamp` <type>,
  PRIMARY KEY (`User_ID`)
);
```

```
CREATE TABLE `AscEditor` (
  `AscEditor_ID` <type>,
  `User_ID` <type>,
  `AscNumPublished` <type>,
  PRIMARY KEY (`AscEditor_ID`),
  KEY `FK` (`User_ID`)
);

CREATE TABLE `Messages-User` (
  `Messages_ID` <type>,
  `User_ID` <type>,
  `Messages_Username` <type>,
  `MessageText` <type>,
  `MesssageTimestamp` <type>,
  KEY `FK` (`Messages_ID`, `User_ID`)
);

CREATE TABLE `Reviewer` (
  `Reviewer_ID` <type>,
  `User_ID` <type>,
  `NumReviewed` <type>,
  PRIMARY KEY (`Reviewer_ID`),
  KEY `FK` (`User_ID`)
);
```

```
);


CREATE TABLE `Messages` (

  `Messages_ID` <type>,

  `Manuscript_ID` <type>,

  PRIMARY KEY (`Messages_ID`),

  KEY `FK` (`Manuscript_ID`)

);


CREATE TABLE `SysAdmin` (

  `SysAdmin_ID` <type>,

  `User_ID` <type>,

  PRIMARY KEY (`SysAdmin_ID`),

  KEY `FK` (`User_ID`)

);


CREATE TABLE `Publication` (

  `Publication_ID` <type>,

  `VolumeNum` <type>,

  `IssueNum` <type>,

  `DatePublished` <type>,

  `DOI` <type>,

  PRIMARY KEY (`Publication_ID`)

);
```

```
CREATE TABLE `Author` (
  `Author_ID` <type>,
  `User_ID` <type>,
  `NumSubmitted` <type>,
  PRIMARY KEY (`Author_ID`),
  KEY `FK` (`User_ID`)
);

CREATE TABLE `Manuscript` (
  `Manuscript_ID` <type>,
  `Editor_ID` <type>,
  `AscEditor_ID` <type>,
  `Author_ID` <type>,
  `Reviewer_ID` <type>,
  `Status` <type>,
  `StatusTimestamp` <type>,
  `ManuscriptTitle` <type>,
  `Keywords` <type>,
  `DateSubmitted` <type>,
  `Reviewers` <type>,
  PRIMARY KEY (`Manuscript_ID`),
  KEY `FK` (`Editor_ID`, `AscEditor_ID`, `Author_ID`, `Reviewer_ID`)
);
```

```
CREATE TABLE `Editor` (
  `Editor_ID` <type>,
  `User_ID` <type>,
  `NumPublished` <type>,
  PRIMARY KEY (`Editor_ID`),
  KEY `FK` (`User_ID`)
);
```

5. References

Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. *Asian Journal of Research in Computer Science*, *4*(2), 1–10. https://doi.org/10.9734/ajrcos/2019/v4i230108

Dutta, P., & Dutta, P. (2019). Comparative Study of Cloud Services Offered by Amazon, Microsoft and Google. *International Journal of Trend in Scientific Research and Development*, *Volume-3*(Issue-3), 981–985. https://doi.org/10.31142/ijtsrd23170

Gargouri, Y., Larivière, V., Gingras, Y., Carr, L., & Harnad, S. (2012). *Green and Gold Open Access Percentages and Growth, by Discipline*. *June 2012*. http://arxiv.org/abs/1206.3664

K, M., LAXMAIAH, M., & SHARMA, Y. K. (2019). a Comparative Study on Google App Engine Amazon Web Services and Microsoft Windows Azure. *International Journal of Computer Engineering & Technology*, *10*(1), 54–60. https://doi.org/10.34218/ijcet.10.1.2019.007

Lederman, N. G., & Lederman, J. S. (2017). The future of peer review. *Journal of Science Teacher Education*, *28*(3), 219–221. https://doi.org/10.1080/1046560X.2017.1319163

Madhuri, T., & Sowjanya, P. (2007). Microsoft Azure v/s Amazon AWS Cloud Services: A Comparative Study. *International Journal of Innovative Research in Science, Engineering and Technology*, *5*(3), 3904–3908. https://doi.org/10.15680/IJIRSET.2016.0503098

Mukherjee, S. (2019). Benefits of AWS in Modern Cloud. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3415956

Park, M., & Seo, T. S. (2016). Creating a national open access journal system: The Korean journal publishing service. *Journal of Scholarly Publishing*, *48*(1), 53–67. https://doi.org/10.3138/jsp.48.1.53

Sahatqija, K., Ajdari, J., Zenuni, X., Raufi, B., & Ismaili, F. (2018). Comparison between relational and NOSQL databases. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, *May*, 216–221. https://doi.org/10.23919/MIPRO.2018.8400041

Tracz, V., & Lawrence, R. (2016). Towards an open science publishing platform. *F1000Research*, *5*, 1–9. https://doi.org/10.12688/f1000research.7968.1

Wicherts, J. M. (2016). Peer review quality and transparency of the peer-review process in open access and subscription journals. *PLoS ONE*, *11*(1). https://doi.org/10.1371/journal.pone.0147913

Pricing Calculator | Microsoft Azure. (n.d.). Retrieved July 6, 2020, from https://azure.microsoft.com/en-us/pricing/calculator/

AWS Pricing Calculator. (n.d.). Retrieved July 6, 2020, from https://calculator.aws/#/

Google Cloud Platform Pricing Calculator. (n.d.). Retrieved July 6, 2020, from https://cloud.google.com/products/calculator