Final Project Documentation for Project Name

**Prepared by:** Project Manager **Date:** 28th August 2025

**I.** Project description

The purpose of this document is to present a detailed description of the Web Publishing System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

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.

**II.** Project proposal

## Statement of work

I own a small project management company specialized in web development with its web developers, hardware resources such as computers fully customizable for web development requirements, meeting rooms to welcome clients requesting web applications.

The Web Application Publishing System requested from the client which will be a University accomplishes the following:

1. Authors can upload their articles

2. Readers can search for articles and read them

3. Reviewers can examine the uploaded articles (Active Articles) and can recommend the approval of articles or to request modifications to be made on the article.

4. Editors who act as admins of the system, they receive the uploaded articles, send articles to be reviewed and with the reviewers recommendation he has the final decision to publish the article on the website or not or send modifications to be made on the article to the Author.

5. A database for all the articles on the system

6. Article Manager is a page where the Active Articles are uploaded and still under reviewing, this page is only accessed by the Editor

7. Online Journal is the public page where Authors upload their articles, reviewers review an article, readers read published articles

8. User Friendly Web Application

9. Operational work & Maintenance to be done after delivering the final product

**III.** Original and revised contract information and client acceptance documents

Project Acceptance

|  |  |
| --- | --- |
| Date | 28th August 2025 |
| Project Name | Web Application Publishing System |
| Client | University |
| Project Sponsor | None |
| Project Manager | Ali Alasmar |

# COMPLETE ACCEPTANCE CRITERIA

1. Functional Requirements mentioned met
2. Security requirements mentioned met
3. User Friendly web application
4. Web Application fully tested to prevent bugs
5. Web Application Scalability can hold 1000-2000 users.
6. Web Application stability guaranteed as possible
7. Web Application compatible with most used browsers (chrome, safari, edge, Firefox)
8. If Web Application crashes it can start up again and recover the state where the user of the website was on.

# PROJECT EVALUATION

On behalf of the University, the individual signing below acknowledges that he or she has reviewed the Phase Acceptance Documentation and has verified that all project deliverables meet the project specifications and requirements. The University acknowledges that there are no unfulfilled obligations remaining. Further, the individual signing below confirms that, he or she, or an authorized agent, has reviewed each of the project deliverables and found each one to either meet or exceed all quality requirements.

# TRANSITION & HAND-OVER

The University acknowledges that the hand-over of the project is complete. For software development projects, this means that the new system is in production and operating properly, that the full source code is in possession of the agreed upon owner, that all system documentation has been delivered to the University.

PERMISSION TO CLOSE PROJECT

By signing below, the University provides the project manager with authorization to perform all project closing activities including releasing the project team.

Date: 28th August 2025

By signing below, I Omar Khaled , in my capacity

as Representative for University , for and on

behalf of University , formally

accept the project described above. I warrant that I have the authority to

accept the project on

behalf of University .

*(Insert Name of Buying Organization)*

*Web Development Company*

Organization Name

By: Ali Alasmar Signature

**IV.** Original and revised project plans and schedules (WBS, Gantt charts and network diagrams, cost estimates)

* **WBS**

1.0 Project Management

1.1 Requirements Analysis

1.1.1 Functional Requirements

1.1.2 Non-Functional Requirements

1.1.2.1 Security Requirements

1.1.2.2 Scalability Requirements

1.1.2.3 Stability Requirements

1.1.2.4 Compatibility Requirements

1.2 Analysis and Design

1.2.1 Acceptance Criteria

1.2.2 Scenarios (Options & Recommendations)

1.2.3 Competitors Analysis

1.2.4 Stakeholder Analysis

1.2.5 Financials Analysis

1.2.6 Functional Analysis

1.2.7 Potential Risks

2.0 Software Development

2.1 Article Manager Page

2.1.1 Beta Version Page

2.1.1.1 Add Author

2.1.1.2 Add Reviewer

2.1.1.3 Update Person

2.1.1.4 Publish Article

2.1.2 Stable Version Page

2.1.2.1 Remove Article

2.1.2.2 Check Status

2.1.2.3 Update Article Status

2.1.2.4 Enter Communication

2.1.2.5 Send Communication

2.1.2.6 Assign Reviewer

2.1.2.7 Adding Database

2.1.2.8 Unit Testing the Page

2.2 Online Journal Page

2.2.1 Beta Version Page

2.2.1.1 Login System

2.2.1.2 Reader Searches an Article

2.2.2 Stable Version Page

2.2.2.1 Author Submits an Article

2.2.2.2 Reviewer Submits a Review of an Article

2.2.2.3 Connect to Database

2.2.2.3 Unit Testing the Page

2.3 Website Final Version

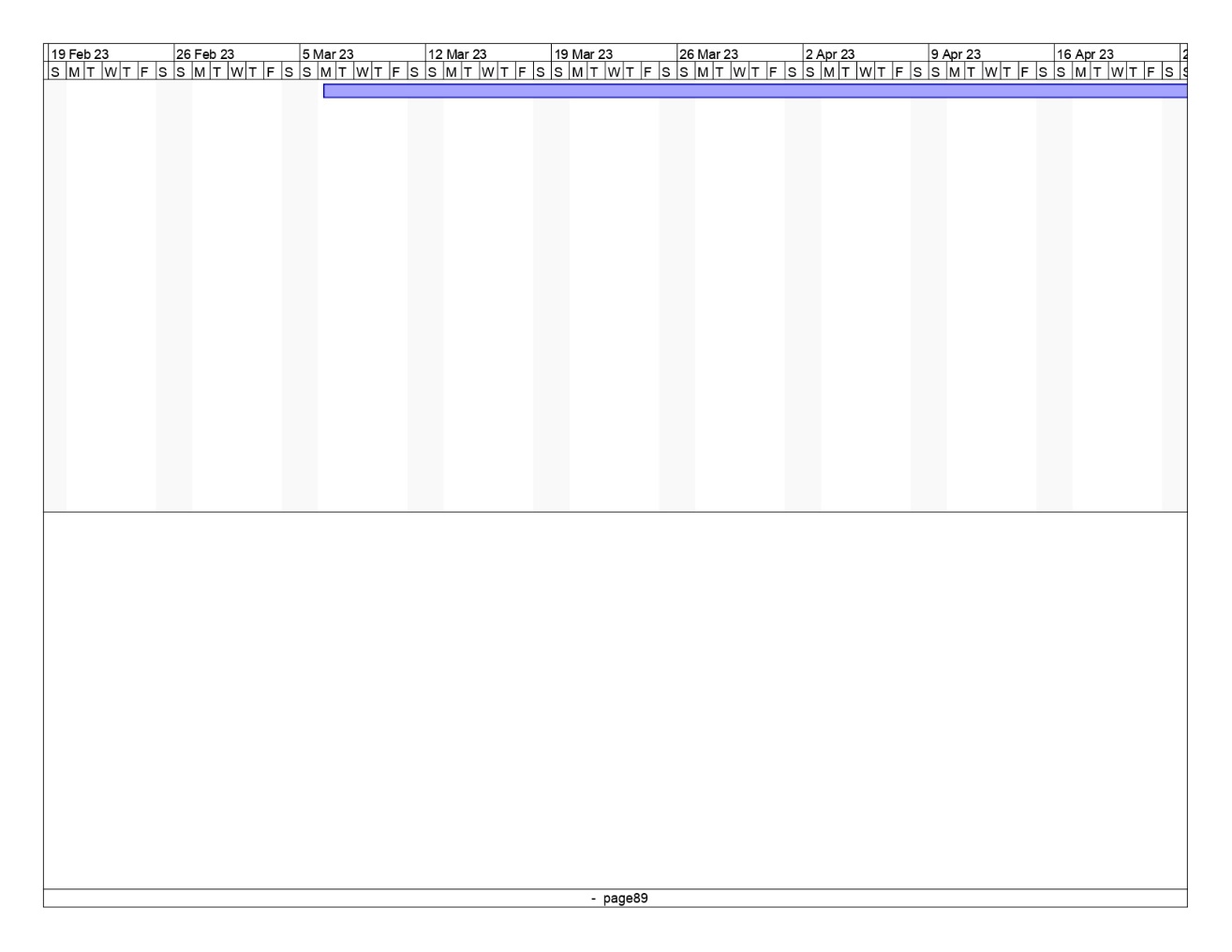
2.3.1 Pages Integration

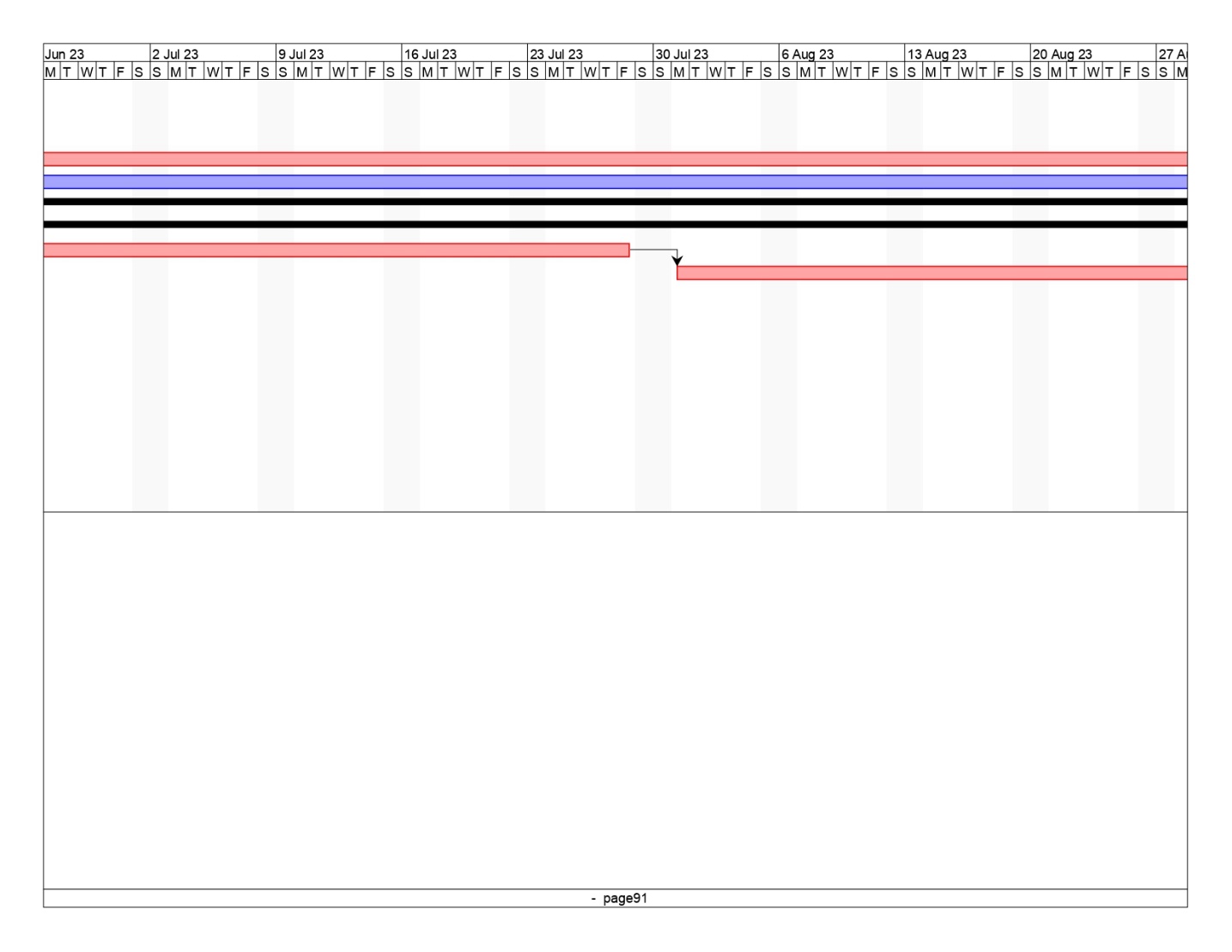
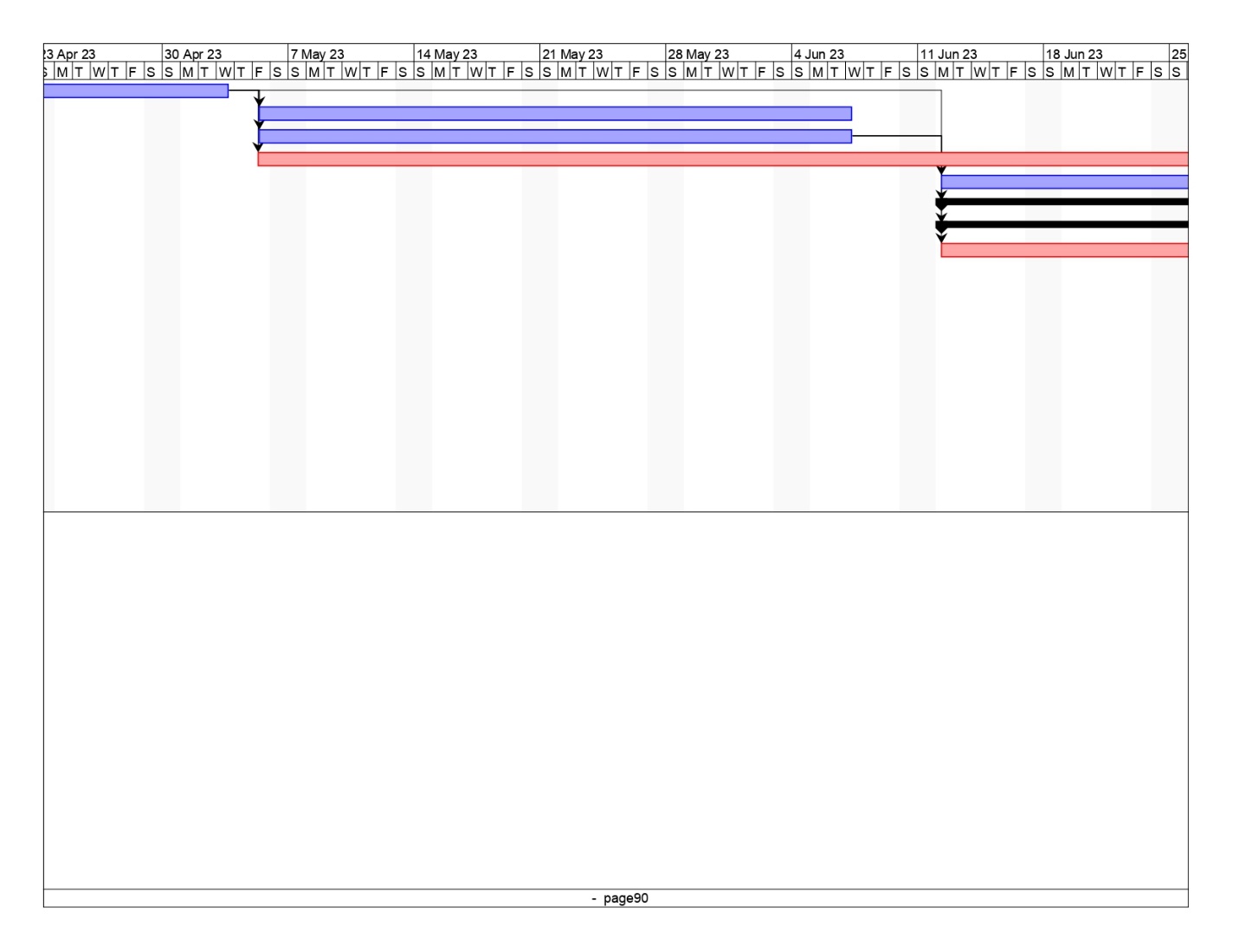
2.3.2 Integration Testing

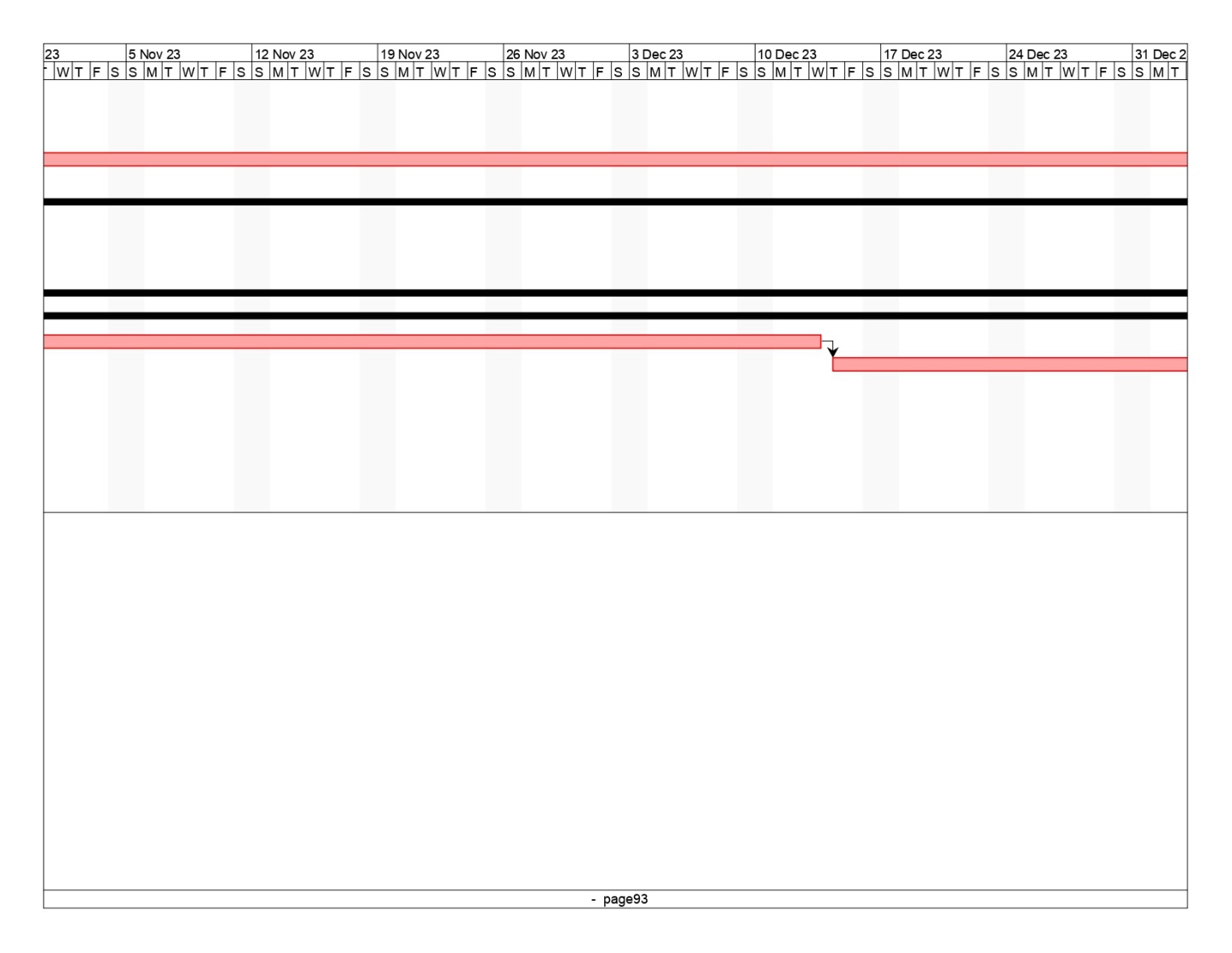
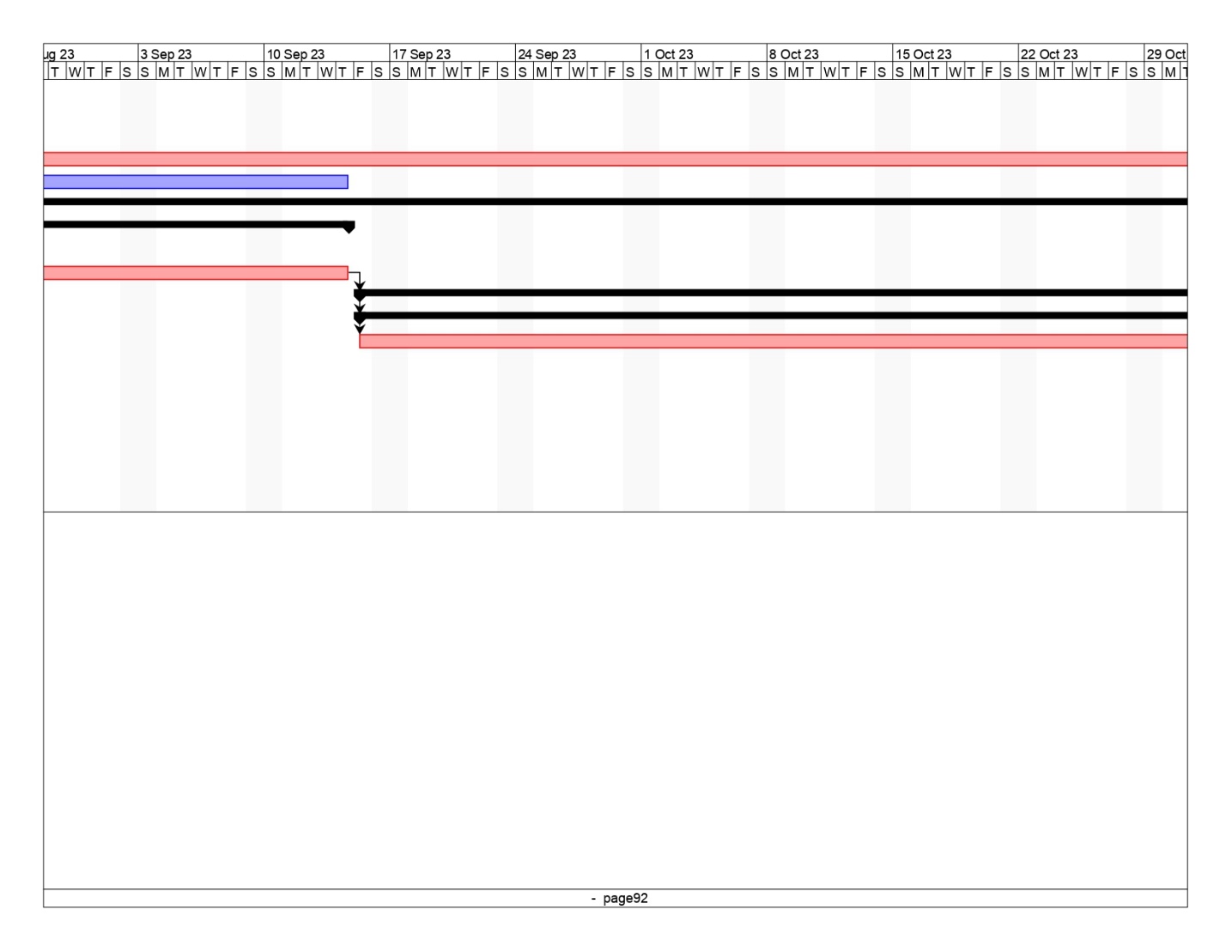
* **Gantt Chart**

**Table

Description automatically generated**

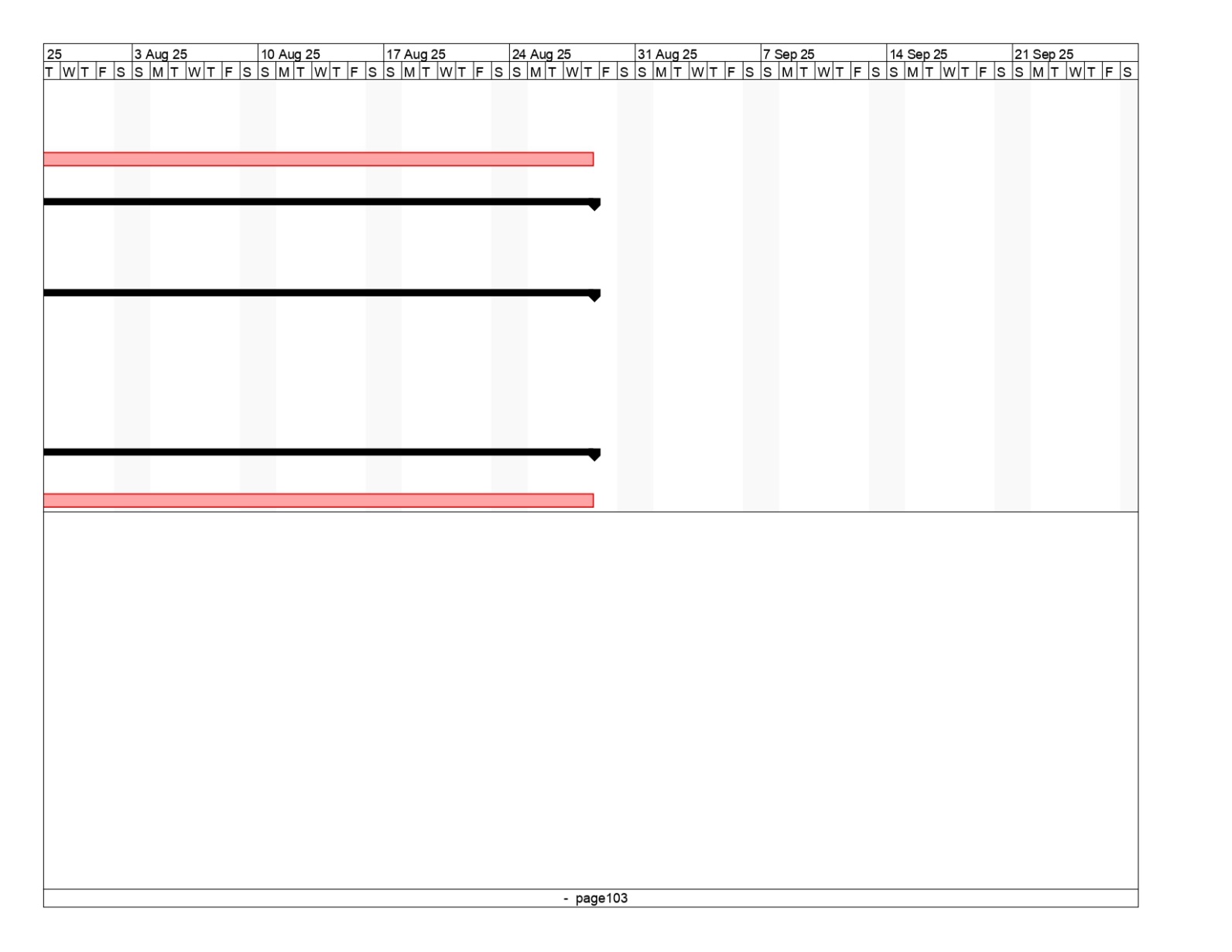
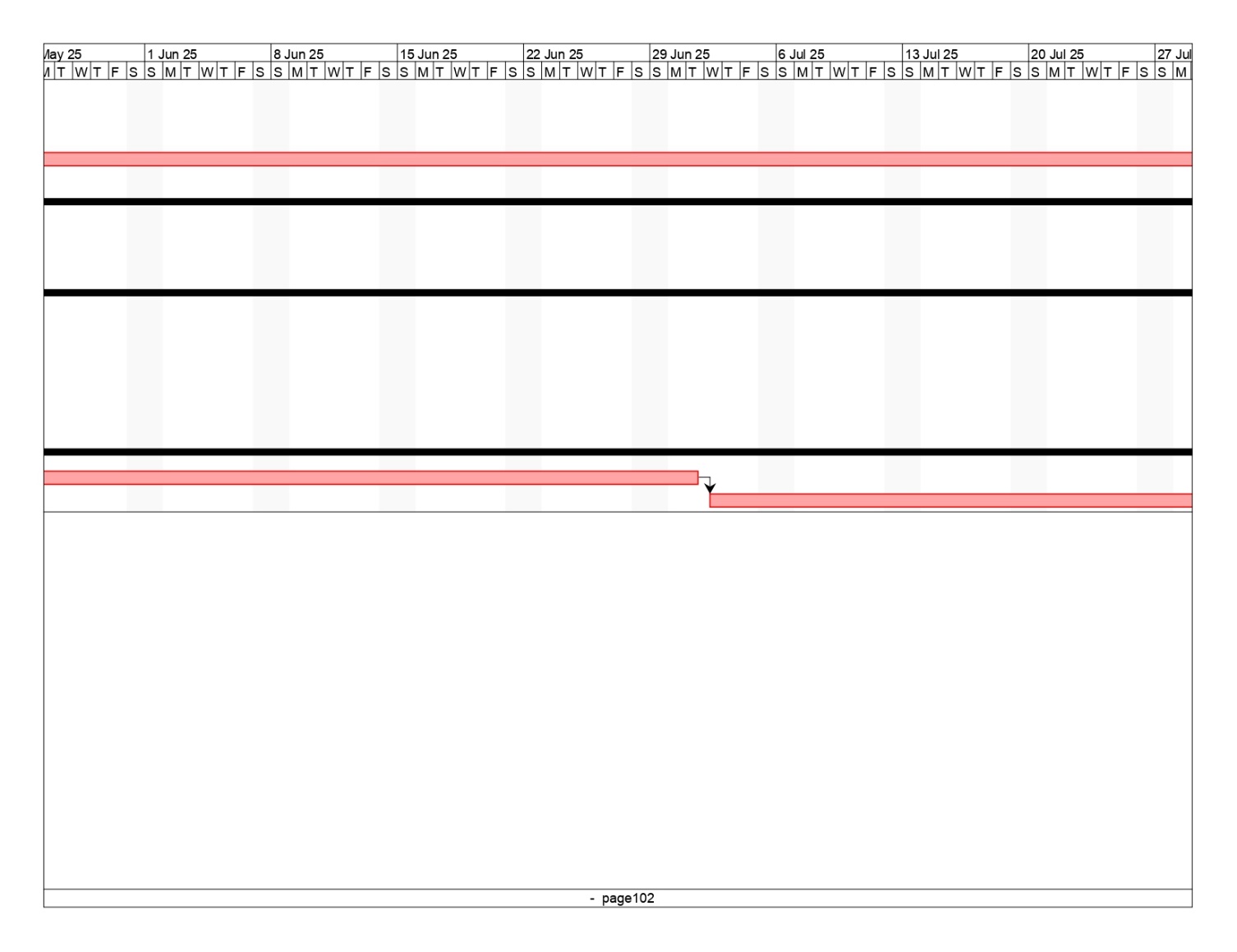
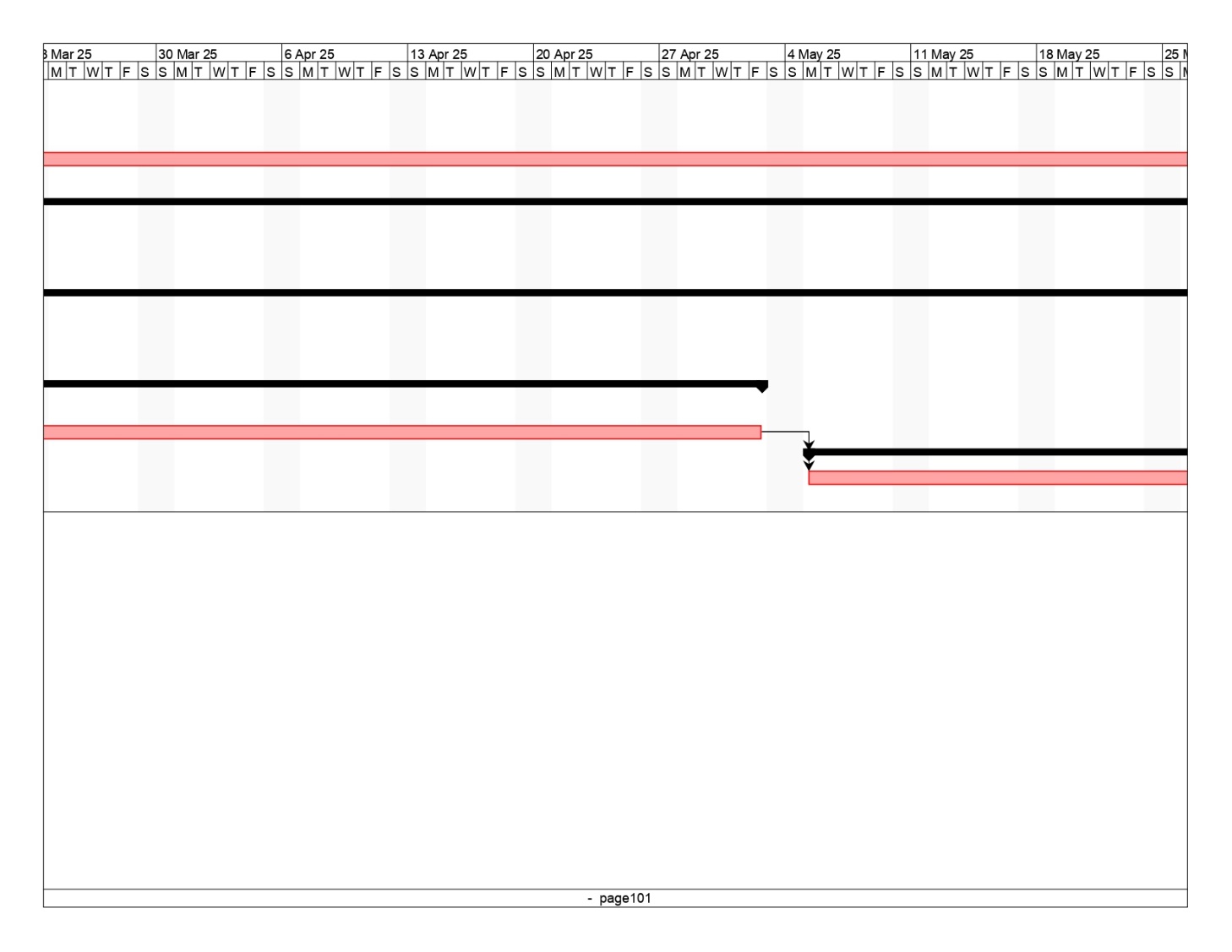
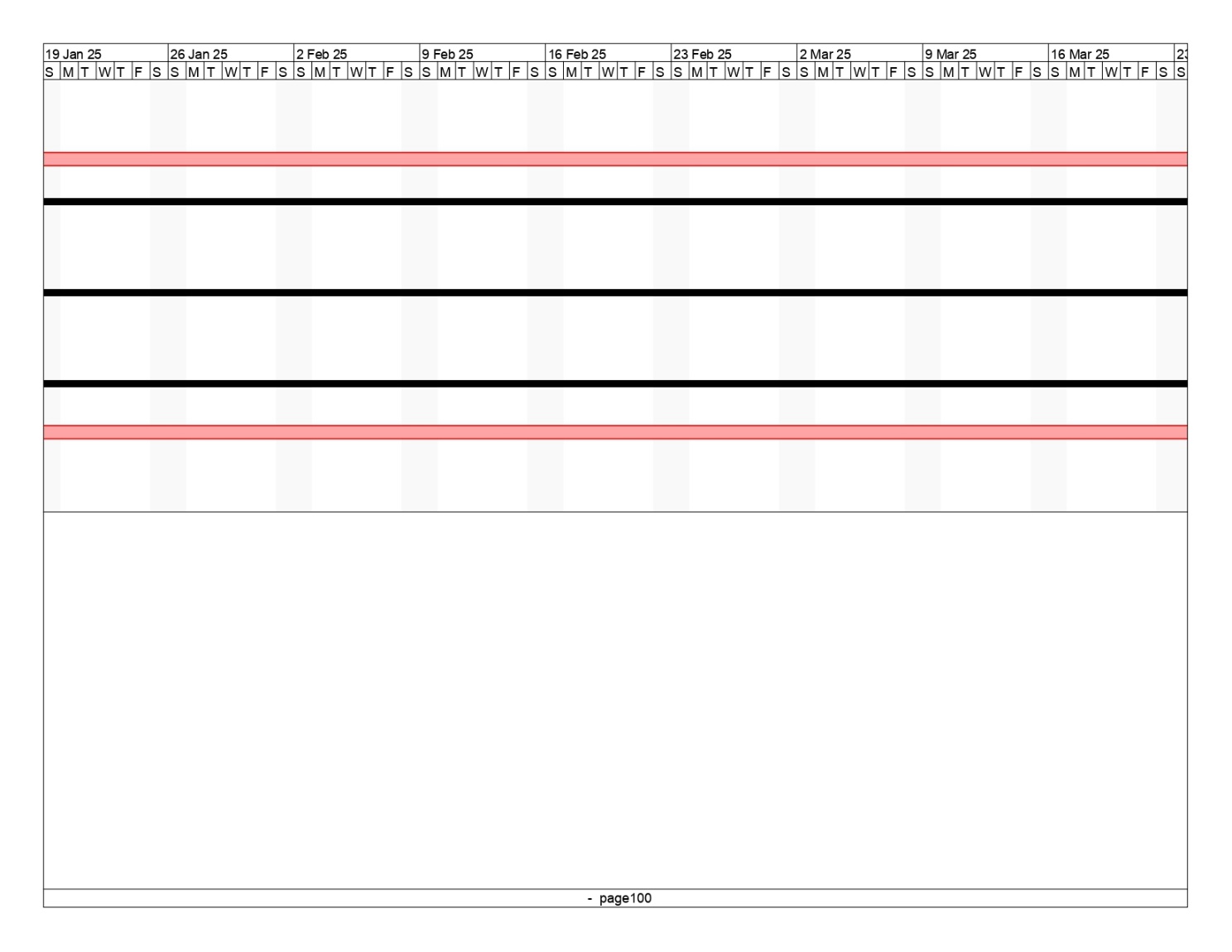
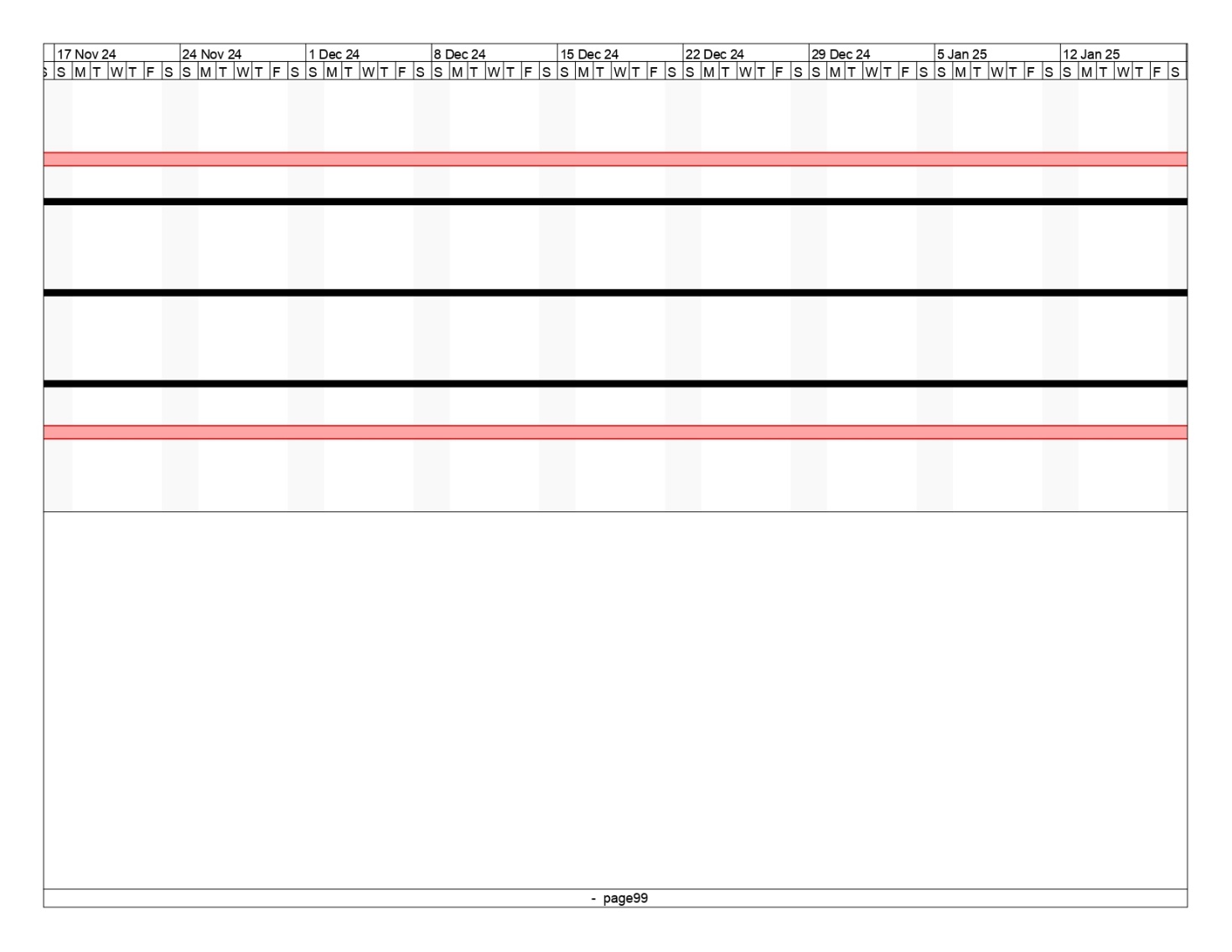
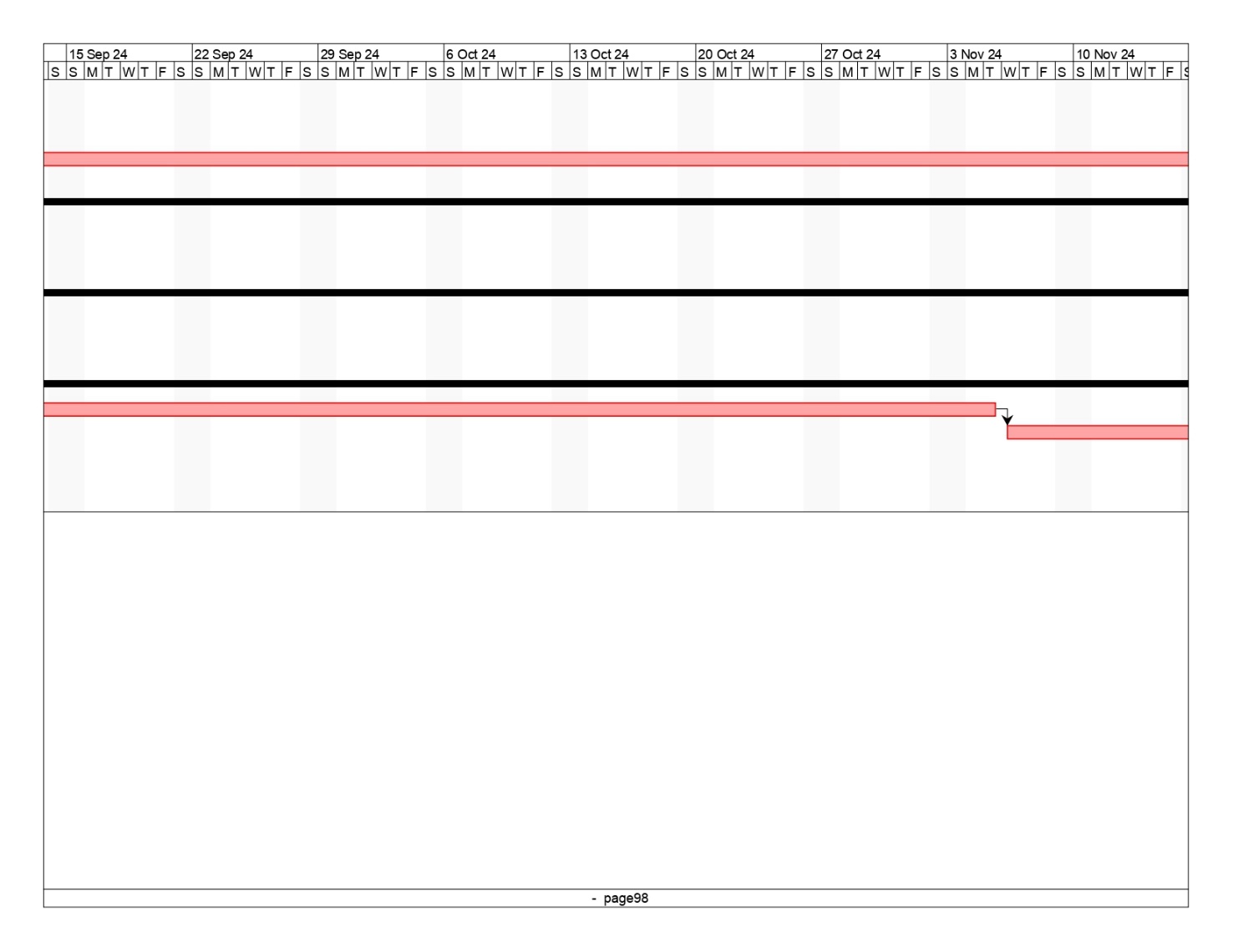
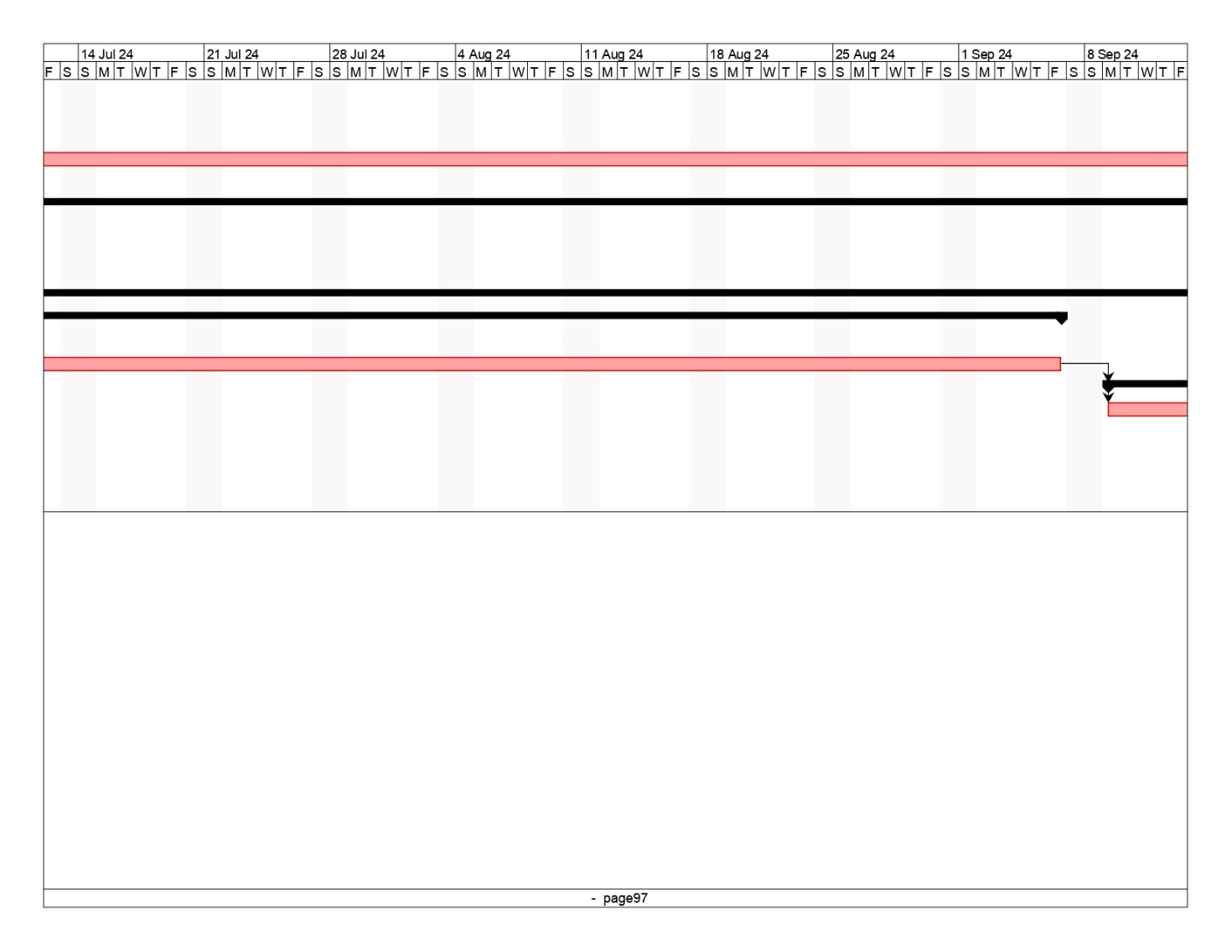
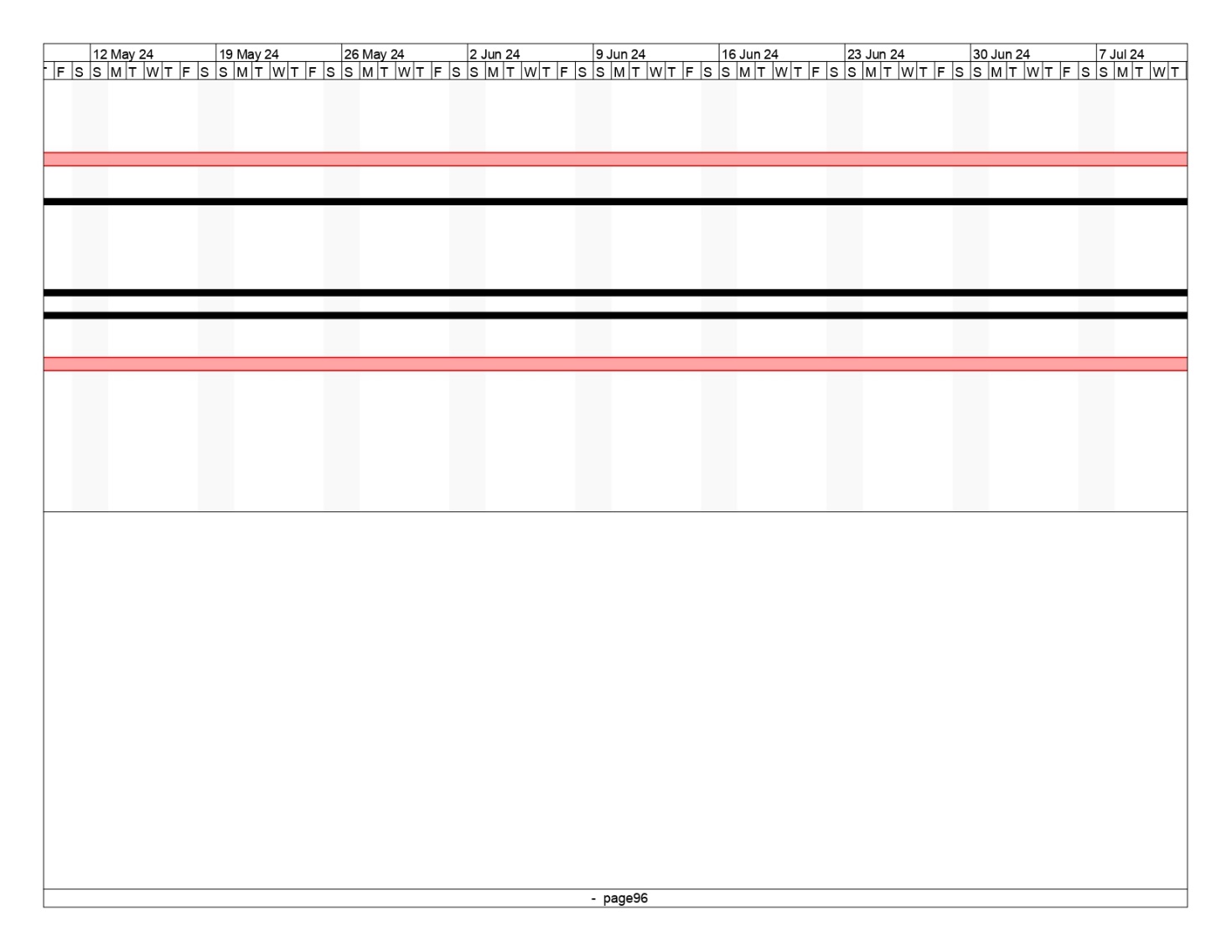
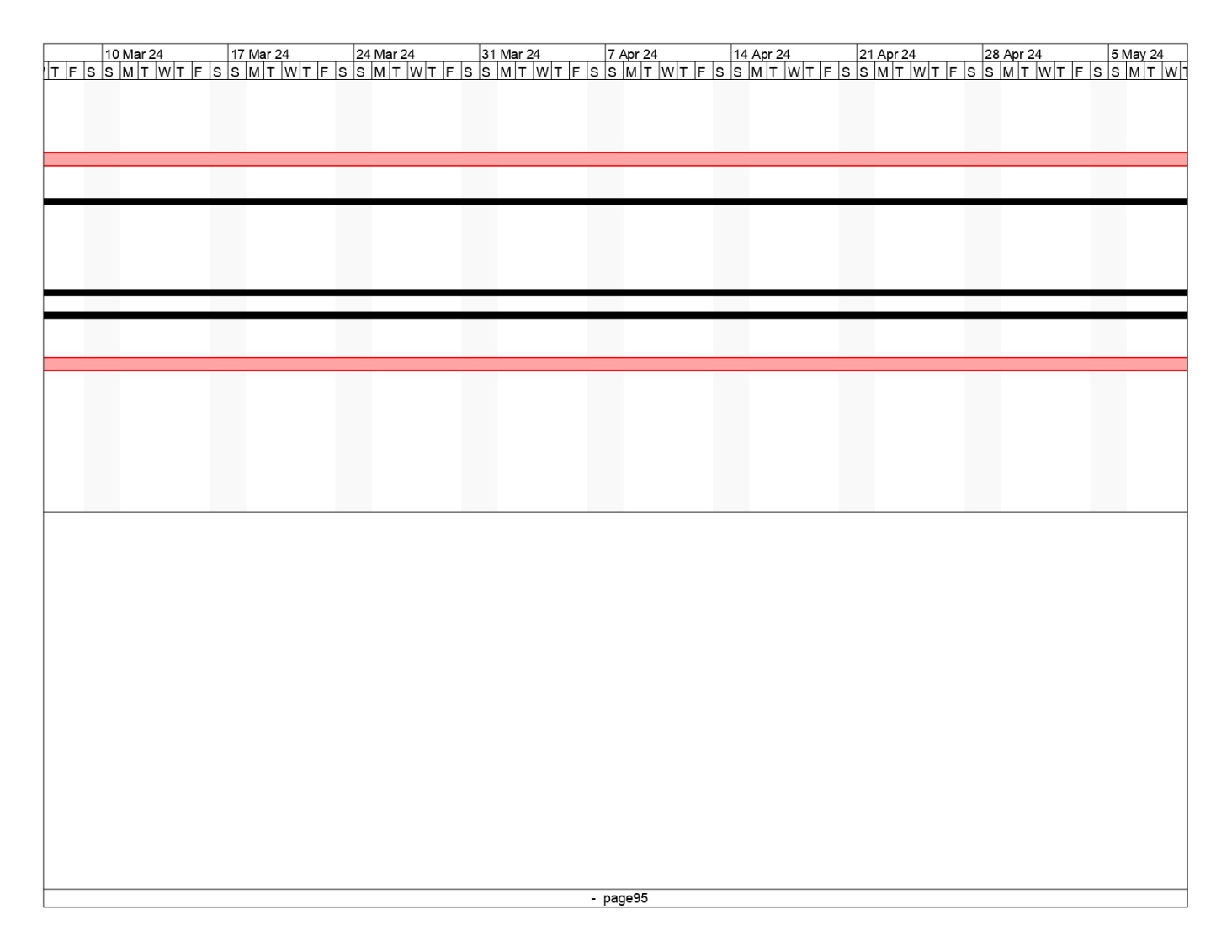
****

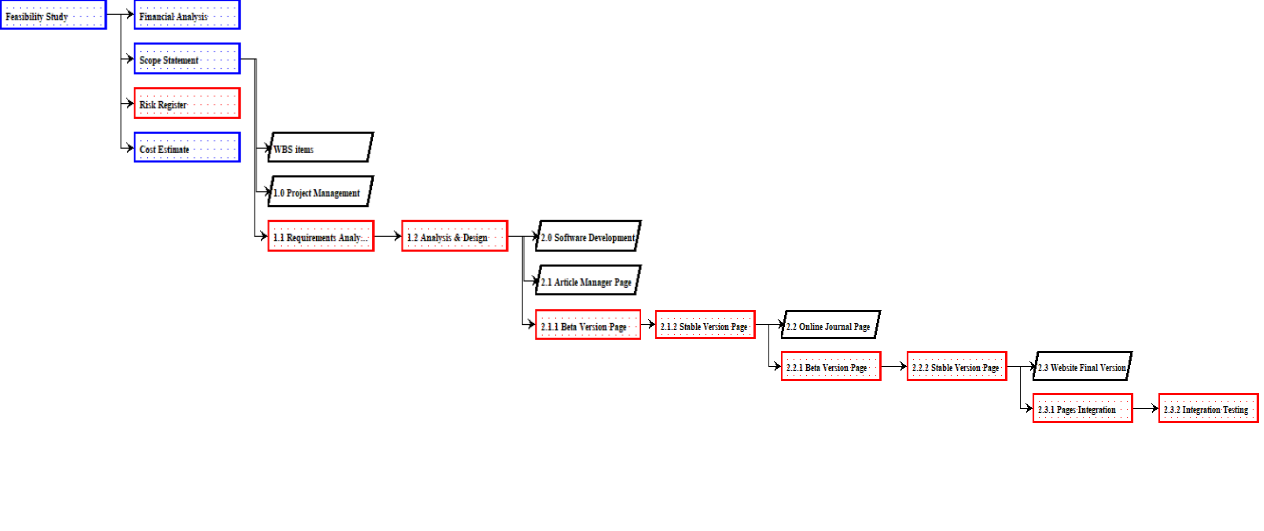
****

****

**A picture containing chart

Description automatically generated**

****

* **Network Diagram**
* **Graphical user interface, application, table, Excel

  Description automatically generatedTable

  Description automatically generatedCost Estimate**

**V.** Design documents

**Figure 1 - System Environment**



Web Publishing System

HS DB

Article Manager

Online Journal

Author

Reader

Editor

Reviewer

The Web Publishing System has four active actors and one cooperating system.



The Author, Reader, or Reviewer accesses the Online Journal through the Internet.

communication with the system is through email. The Editor accesses the entire system directly. There is a

Any Author or Reviewer

link to the (existing) Historical Society.

<< The division of the Web Publishing System into two component parts, the Online Journal and the Article Manager, is an example of using domain classes to make an explanation clearer. >>

* 1. **Functional Requirements Specification**

This section outlines the use cases for each of the active readers separately. The reader, the author and the reviewer have only one use case apiece while the editor is main actor in this system.

## Reader Use Case

Use case: **Search Article**

## Diagram:



Search Article

Reader

**Brief Description** The Reader machine.

accesses the Online Journal Website, searches for an article and downloads

it to his/her

## 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.

1. The system

displays the choices t

1. The Reader

selects the article

1. The system

presents the abstract of the article

1. The Reader chooses to

download the article.

o the Reader. desired.

to the reader.

1. The system provides the requested article.

**Xref:** Section 3.2.1, Search Article



Rewrite

Review

Active Article

Submit

Publish

**Figure 2 - Article Submission Process**

The *Article Submission Process* state-transition diagram summarizes the use cases listed below.

Editor enters it into the system

submits an article for consideration.

An Author

on the article. Either the a

changes based on the reviews.

The

and assigns it to and which are used by the

or the

declined,

rticle is accepted as written,

Reviewers return their comments,

The

least three reviewers.

Editor sends a copyright form

Author is asked to make some

Editor to make a decision

sends it to at

When that

to the Author.

If it is accepted, possibly after a revision , the the article is

n the

Not shown i

above is the removal of a declined article from the system.

published to the Online Journal.

form is returned,

## Author Use Case

In case of multiple authors, this term refers to the

*principal author*,

with whom all communication is made.

Use case: Submit Article

## Diagram:



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.

1. The System uses the

*sendto* HTML tag to bring up the user’s email system.

1. The Author fills in the

Subject line and attaches the files

as directed and emails them.

1. The System generates and sends an email acknowledgement.

**Xref:** Section 3.2.2, Communicate

* + 1. **Reviewer Use Case** Use case: Submit Review **Diagram:**

Submit Review

Reviewer

## Brief Description

The reviewer submits a review of an article.

## Initial Step-By-Step Description

Before this use case can be initiated, the Reviewer has already connected to the Online Journal Website.

1. The Reviewer chooses the

*Email Editor* button.

1. The System uses the

*sendto* HTML tag to bring up the user’s email system.

1. The Reviewer fills in the

Subject line and attaches the file

as directed and emails it.

1. The System generates and sends an email acknowledgement.

**Xref:** Section 3.2.2, Communicate

## Editor Use Cases

The Editor has the following sets of use cases:



Update Info

Handle Art

Editor

Ck Status

Send Rec

Publish Art

**Figure 3 - Editor Use Cases**

## Update Information use cases

Use case: Update Author

## Diagram:



Update Author

Editor

## Brief Description

The Editor

enters a new Author

or updates information about a current Author.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

1. The Editor selects to *Add/Update Author*.
2. The system presents a choice of adding or updating.
3. The Editor chooses to add or to update.

list of authors to choose from

1. If the Editor is

updating an Author, t

filling in with the information;

else the system presents a blank grid.

he system presents a

and presents a grid

1. The Editor fills in the information and submits the form.
2. The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.3, Add Author; Section 3.2.5 Update Person

Use case: Update Reviewer

## Diagram:



Update Reviewer



Editor

## Brief Description

Hist Soc DB

The Editor enters a new Reviewer or updates information about a current Reviewer.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

* 1. The Editor selects to *Add/Update Reviewer*.
  2. The system presents a choice of adding or updating.
  3. The Editor chooses to add or to update.
  4. The system links to the Historical Society Database.
  5. If the Editor is updating a Reviewer, the system and presents a grid with the information about the Reviewer; else the system presents list of members for the editor to select a Reviewer and presents a grid for the person selected.
  6. The Editor fills in the information and submits the form.
  7. The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.4, Add Reviewer; Section 3.2.5, Update Person

Use case: Update Article

## Diagram:



Update Article

Editor

## Brief Description

The Editor enters information about an existing article.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

1. The Editor selects to *Update Article*.
2. The system presents s list of active articles.
3. The system presents the information about the chosen article.
4. The Editor updates and submits the form.
5. The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.6, Update Article Status

## Handle Article use cases

Use case: Receive Article

## Diagram:



Receive Article

Editor

## Brief Description

The Editor enters a new or revised article into the system.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager and has a file containing the article available.

1. The Editor selects to *Receive Article*.
2. The system presents a choice of entering a new article or updating an existing article.
3. The Editor chooses to add or to update.
4. If the Editor is updating an article, the system presents a list of articles to choose from and presents a grid for filling with the information; else the system presents a blank grid.
5. The Editor fills in the information and submits the form.
6. The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.7, Enter Communication

Use case: Assign Reviewer

This use case extends the *Update Article* use case.

## Diagram:



Assign Reviewer



Editor

## Brief Description

The Editor assigns one or more reviewers to an article.

Hist Soc DB

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

1. The Editor selects to *Assign Reviewer*.
2. The system presents a list of Reviewers with their status (see data description is section 3.3 below).
3. The Editor selects a Reviewer.
4. The system verifies that the person is still an active member using the Historical Society Database.
5. The Editor repeats steps 3 and 4 until sufficient reviewers are assigned.
6. The system emails the Reviewers, attaching the article and requesting that they do the review.
7. The system returns the Editor to the *Update Article* use case.

**Xref:** Section 3.2.8, Assign Reviewer

Use case: Receive Review

This use case extends the *Update Article* use case.

## Diagram:



Receive Review

Editor

## Brief Description

The Editor enters a review into the system.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

1. The Editor selects to *Receive Review*.
2. The system presents a grid for filling with the information.
3. The Editor fills in the information and submits the form.
4. The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.7, Enter Communication

## Check Status use case:

Use case: Check Status

## Diagram:



Check Status

Editor

## Brief Description

The Editor checks the status of all active articles.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

1. The Editor selects to *Check Status*.
2. The system returns a scrollable list of all active articles with their status (see data description in section 3.3 below).
3. The system returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.9, Check Status

## Send Recommendation use cases:

Use case: Send Response

This use case extends the *Update Article* use case.

## Diagram:



Send Response

Editor

## Brief Description

The Editor sends a response to an Author.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

1. The Editor selects to *Send Response*.
2. The system calls the email system and puts the Author’s email address in the Recipient line and the name of the article on the subject line.
3. The Editor fills out the email text and sends the message.
4. The system returns the Editor to the Article Manager main page.

**Xref:** Section 3.210, Send Communication

Use case: Send Copyright

This use case extends the *Update Article* use case.

## Diagram:



Send Copyright

Editor

## Brief Description

The Editor sends a copyright form to an Author.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

1. The Editor selects to *Send Copyright*.
2. The system calls the email system and puts the Author’s email address in the Recipient line, the name of the article on the subject line, and attaches the copyright form.
3. The Editor fills out the email text and sends the message.
4. The system returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.10, Send Communication

Use case: Remove Article

This use case extends the *Update Article* use case.

## Diagram:



Remove Article

Editor

## Brief Description

The Editor removes an article from the active category.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

1. The Editor selects to remove an article from the active database.
2. The system provides a list of articles with the status of each.
3. The Editor selects an article for removal.
4. The system removes the article from the active article database and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.12, Remove Article

## Publish Article use case:

Use case: Publish Article

This use case extends the *Update Article* use case.

## Diagram:



Publish Article

Editor

## Brief Description

The Editor transfers an accepted article to the Online Journal.

## Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

1. The Editor selects to *Publish Article*.
2. The system transfers the article to the Online Journal and updates the search information there.
3. The system removes the article from the active article database and returns the Editor to the Article Manager home page.

**VII.** Deliverables

Deliverable 1: A Beta Version of the Article Manager page for the Editors will be delivered in 3 months.

Deliverable 2: Then the Article Manager page (Stable Version) fully functional will be delivered in 4 months with the database connected to it. Testing the Article Manager page will take 2 months after those 4 months.

We will add up 3 months if things get wrong.

So, in total 12 months for this web site page to be delivered

First year Deliverable 1&2

Deliverable 3: Online Journal Page for Readers, Reviewers & Authors will have a Beta version delivered in 2 months.

Deliverable 4: Then the Online Journal Page (Stable Version) fully functional will be delivered in 4 months. Testing the Online Journal Page will take 2 months after those 4 months.

So, in total 8 months for this web site page to be delivered.

Deliverable 5 (Last Deliverable): Then we will integrate the two pages in 2 months, now the system is fully functional. Lastly, we will take extra 2 months to test the whole system so, in total 4 months for the integrated tested system to be delivered.

Second year Deliverable 3&4&5

So, 2 years is needed to deliver the whole web product.

**IX.** Lessons learned reports

|  |
| --- |
| **Prepared by: Project Manager** **Date: 28th August 2025**  **Project Name: Web Application Publishing System**  **Project Sponsor:** None  **Project Manager:** **Ali Alasmar**  **Final Cost: 510,000 Final Benefit: 637,000** |
| 1. Did the project meet scope, time, and cost goals?   Scope: Met perfectly  Time: Late by 2 months  Cost:  From the company we paid 510,000 in costs, but we had 637,000 in return of the Website  So, it was profitable for the company   1. What was the success criteria listed in the project scope statement?   1.Functional Requirements mentioned met  2.Security requirements mentioned met  3.User Friendly web application  4.Web Application fully tested to prevent bugs  5.Web Application can hold 1000-2000 users.  6.Web Application stability guaranteed as possible  7. Web Application compatible with most used browsers (chrome, safari, edge, Firefox)  8.If Web Application crashes it can start up again and recover the state where the user of the website was on.   1. Reflect on whether or not you met the project success criteria.   We meet all acceptance criteria except:  1.Application Stability, developers worked on that  2.Application Scalability, we didn’t work on this because Client mentioned it late (as stated in Risk Register)   1. In terms of managing the project, what were the main lessons your team learned?   1. We should have had a subscription for a reliable Database Service instead of using a free one (as stated in Risk Register)  2.We should have more developers working on projects, overworking the developers was the Biggest Risk (as stated in Risk Register)   1. Describe one example of what went right on this project.   1. Meetings throughout the project was very satisfying to always get feedback on the development.  2. Developers have gained experience on the Potential of Publishing Websites, although it was very hard regarding their small number.  3. One of the things we did very well was testing every deliverable and even having beta releases was very exciting to the Client to see how much we progressed   1. Describe one example of what went wrong on this project.   1. Payback was only in last year  2. NPV is relatively low.  3. Development was on the edge and depending on small number of developers   1. What will you do differently on the next project based on your experience working on this project?   1. I think we should pick a higher paid University Website next time, even if it requires more work.  2. Certainly when I (Project Manager) can hire more developers I will. |

**X.** Copies of all status reports, meeting minutes, change notices, and other written and electronic communications

Meeting Minutes & Reports & Feedbacks

| **Time** | **Item** | **Duration** |
| --- | --- | --- |
| 10th June 2023 | Team Contract Meeting | 1 hour |
| 12th June 2023 | Kick-Off Meeting | 2 hours |
| 13th June 2023 | Analysis & Design Meeting | 2 hours 15 min |
| 15th September 2023 | 1st Day of Development | 1 hour |
| 10th October 2023 | Beta Version Article Manager Page Deliverable Feedback 1 | 3 hours |
| 14th December 2023 | Beta Version Article Manager Page Deliverable Feedback 2 | 3 hours |
| 5th May 2024 | Stable Version Article Manager Page Deliverable Feedback 1 | 2 hours 45 min |
| 7th September 2024 | Stable Version Article Manager Page Deliverable Feedback 2 | 2 hours 45 min |
| 10th October 2024 | Beta Version Online Journal Page Deliverable Feedback 1 | 3 hours |
| 6th November 2024 | Beta Version Online Journal Page Deliverable Feedback 2 | 3 hours |
| 5th February 2025 | Stable Version Online Journal Page Deliverable Feedback 1 | 2 hours 45 min |
| 4th May 2025 | Stable Version Online Journal Page Deliverable Feedback 2 | 2 hours 45 min |
| 15th May 2025 | Website Final Version Feedback 1 | 3 hours |
| 15th July 2025 | Website Final Version Feedback 2 | 3 hours 40 min |
| 15th October 2025 | Website Final Version Feedback 3 | 3 hours 30 min |
| 1st September 2025 | Website Final Version Feedback 4 | 3 hours |
| 3rd September 2025 | Project Closing | 2 hours |
|  |  | **TOTAL DURATION**  **44 HOURS**  **25 MIN** |