COMP 693: INDUSTRY PROJECT PROPOSAL

Lincoln University
Information Technology Service (ITS)
Change Advisory Board Upgrade

(Industry Project)

Submitted by

Kithusan Albert

8000300

Company: Lincoln University Information Technology Services

Company Address: Lincoln 7647, Christchurch, New Zealand

Mentor: Brandon Kennedy

Mentor email: Brandon.Kennedy@lincoln.ac.nz

Date: 18 July 2024

1. Background

1.1 Overview

Lincoln University is a public university in Lincoln, New Zealand that was formed in 1990 and is the oldest agricultural teaching institution in the southern hemisphere.

At Lincoln University, the Information Technology Service (ITS) department is responsible for the management and maintenance of the university's technological infrastructure and services.

The ITS department use a change management system called GAS. This system is to ensure the smooth implementation of technology changes and the system helps in the tracking and documenting of changes to critical and non- critical services, obtaining necessary approval from relevant stakeholders and maintaining overall accountability of changes. GAS also enables ITS department to record all modifications made to a technology and test and review changes at any stage.

This project is to create a prototype Change Management System that utilizes existing approved technologies at Lincoln University Information Technology Service department. The new system will focus on streamlining approval process and improving user experience through features like email approvals and a user-friendly UX design.

1.2 Problem

The Lincoln University Information Technology Service department is still using an outdated change management system that is causing inefficiencies and frustration for its users. This outdated system has several issues:

1. Lacks modern features that can make the process more efficient:
   Requires users to visit a separate portal from the approval notification email to approve or decline a change request. A modern feature that could make the process easier and faster
2. Outdated user experience and design:
   The user interface design is outdated and clunky, causing confusion for new users
3. Built on outdated technology:
   The technology of the current system is outdated and at the end of its lifecycle.

1.3 Importance of addressing issues

The issues are important to be addressed for Lincoln University Information Technology Service department because:

1. **Efficiency:** The outdated system creates inefficient process and impact productivity, leading to wasted time and potential delays in critical IT changes.
2. **User Experience:** A modern system will be overall a better user-friendly experience for staff.
3. **Integration:** A modern system will seamlessly integrate with existing Lincoln University's Microsoft IT infrastructure and technology.
4. **Advanced features:** Provide additional features such as reporting and change request metrics to support management and IT staff.

1.4 Project Team

The project owner is Andrew Frapwell, the IT Service Manager and owner of the Lincoln University change management system. He will be the main stakeholder.

Email: andrew.frapwell@lincoln.ac.nz

Phone: +6434230118

The project supervisor is Brandon Kennedy, who is the ITS Operations Team Leader. Who has specialist experience in various technologies and will provide guidance on technology and technical decisions.

Email: brandon.kennedy@lincoln.ac.nz

Phone: +6434230116

2. Goal and requirements

2.1 Metrics of success

The primary goal of this project is to upgrade Lincoln University ITS's legacy change management system to a modern, efficient system that enhances user experience and improves the overall change management process.

Functionality:

- **Deliver a functional change management system prototype:** Create a new efficient and improved system to the current change management system and workflows.
- **Integrate with Microsoft Technology:** Integrate Microsoft connectors (Lincoln University Azure Active Directory) and implement Role-Based Access Control.

Alignment with Lincoln ITS Standards:

- **Leverage existing approved technologies:** The system should utilize technologies already vetted and approved by Lincoln ITS for seamless integration and ongoing support.
- **Maintainability by current staff:** The chosen technologies and development approach should be within the skillset of existing Lincoln ITS staff, ensuring efficient maintenance and future enhancements.

User Adoption and Efficiency:

- **Positive stakeholder feedback:** Stakeholders should find the new system to be efficient and significantly improve their change management workflows.
- **Streamlined workflows:** The new system should automate tasks, improve data flow, and reduce processing times, leading to more efficient change management processes.

Testing:

- **Unit Testing:** Perform thorough unit testing during development to ensure all functionalities work correctly without errors.
- **Final Testing:** Perform a final functionality test after deployment to ensure the system works as intended in the live environment.

3. Method

3.1 Overview

Upgrading Lincoln University's ITS legacy change management system is structured using a phased approach. This project will incorporate elements of waterfall project management methodology and Agile development methodology. Actively including stakeholder involvement throughout the project phases and continuously gather feedback on development and decisions made at each phase.

Phase 1: Requirements Gathering

1. Assessing current system workflows and user needs and identifying areas for improvement. Identifying areas within the change management policy to be translated to other functionalities that does not exist currently.
2. Gap Analysis interview with stakeholder and end-users to identify features that can be included in the new system.

Phase 2: Design

1. Choosing the types of technology to use with the guidance of the project supervisor.
2. Designing the user interface, system design and database.
3. Planning the integration of necessary Microsoft connectors, and for implementing Role-Based Access Control (RBAC) using Azure Active Directory.

Phase 3: Development

1. Development of the new system based on the design specifications. This phase includes integrating Microsoft connectors, implementing RBAC, and ensuring the system components work together seamlessly.
2. Iterative feedback from supervisor and key stakeholder will be conducted throughout the development process.

Phase 4: Testing and Deployment

1. Unit testing during development as well as a test plan to ensure all functionalities work correctly without errors.
2. Phase 4 will also include gathering feedback from stakeholders for refinements and deployment in a controlled environment.

Phase 5: Documentation and Training

1. Deployment and troubleshooting issues related to live environment
2. ITS staff training session and a simple user guide.

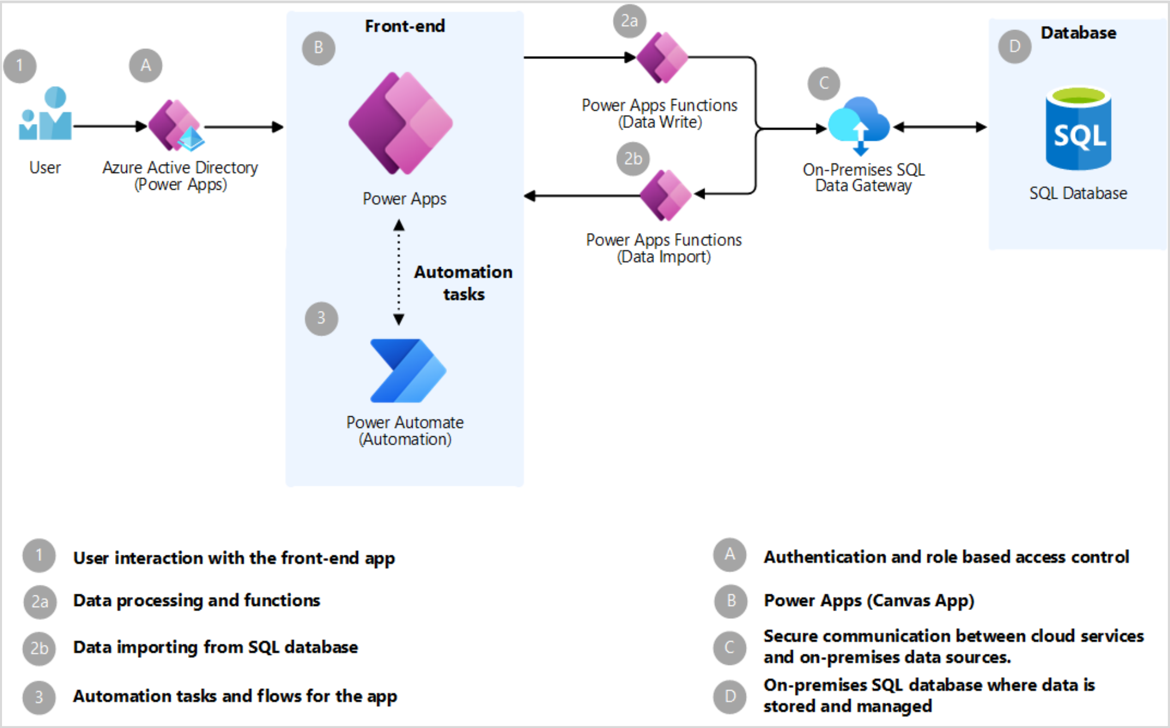
3.2 Design

The following diagram shows the system technology architecture for the solution, which utilizes Microsoft Power Platform as its core technology.

The Power Platform technology approach is chosen due Lincoln University's existing investments in the platform and this approach ensures that the solution is a secure, scalable and easily maintainable solution.

Additionally, a SQL database schema will be developed from scratch. This application is intended to fully replace the old system.

Diagram 1: Change Advisory Board System Architecture:


1
User
A
Azure Active Directory (Power Apps)
Front-end
B
Power Apps
Automation tasks
3
Power Automate (Automation)
2a
Power Apps Functions (Data Write)
2b
Power Apps Functions (Data Import)
C
On-Premises SQL Data Gateway
D
Database
SQL
SQL Database
1 User interaction with the front-end app
2a Data processing and functions
2b Data importing from SQL database
3 Automation tasks and flows for the app
A Authentication and role based access control
B Power Apps (Canvas App)
C Secure communication between cloud services and on-premises data sources.
D On-premises SQL database where data is stored and managed

3.3 New and Advanced Skills to be Gained

Skills already learned in the MAC to be used for this project:

1. Designing the database schemas for change management system and writing complex SQL queries.
2. Maintaining best programing practices during development.
3. Understanding of web development, system design principles and testing methodologies.
4. Understanding of requirements gathering and analysis.

New skills to be learned for this project:

1. Power Apps Platform Development: Creating custom app, designing and creating intuitive user experience (UX) and implementing complex business logic.
2. Power Automate Platform: Building automation workflows and integrating with other Microsoft products.
3. SQL Integrations with Power Apps: Setting up On-Premises Data Gateway between cloud app and on-premises data. Writing database read and write queries. Troubleshooting connectivity and query issues.
4. Role-based Access Control: Implementing role-based access control within the Application. Managing application access control in a cloud environment.
5. Design: Designing intuitive user experience (UX) for business use case.
6. Process: Understanding existing business process and translating into application workflow and enhancements.

3.4 Risks and Challenges

Potential risks and challenges the project might face in upgrading Lincoln University's change management system.

1. **Balancing project workload with course work and other responsibilities**

   Risk Mitigation:

   - Create a detail project timeline and set realistic project milestones.
   - Maintain proactive communication about delays and challenges with industry supervisor and stakeholder.
2. **Resistance to system change and hindering project adoption**

   Risk Mitigation:

   - Involve end-users in the project early to gather requirements and key-stakeholders involvement throughout the project.
   - Provide training and support for the new system.
3. **Limited and slow development with Power Platform due the steep learning curves of learning a new technology.**

   Risk Mitigation:

   - Allocating time for self-study and to get familiar with the technology.
   - Communicate and seek help from inhouse Power Platform expert.

4. **Complexity of integrating on-premises SQL database with cloud-based application.**

   Risk Mitigation:

   - Allocating time for self-study of the On-premises Data Gateway and its functionality.
   - Communicate and seek with industry supervisor for help when required.
5. **Technical issues with new technology**

   Risk Mitigation:

   - Involve end-users and stakeholder in the project early to gather requirements.
   - Communicate and seek with industry supervisor for help when required.

6. **Accidental deletion or overwriting of project source code**

   Risk Mitigation:

   - Use best development practices. Create development, testing and production environments
   - Regular backup of project files and have option to revert to last known good state of developed application.

7. **Technical issues working with multiple technologies**

   Risk Mitigation:

   - Proactively research compatibility and known issues of various technology being used before starting development stage
   - Use best development practices recommended by Microsoft and use latest documentations

8. **Unforeseen personal circumstances that can create delays in meeting project deadlines.**

   Risk Mitigation:

   - Use effective time management and prioritize big tasks that should be completed at the start of each phase.
   - Regular reviews of project progress to identify any potential delays early.
   - Communicate and seek support from industry supervisor

3.5 Implementation Plan

Requirements Gathering

1. Conduct stakeholder interview to gather success criteria and document system requirements
2. Analyze current system workflow and business processes.
3. Identify areas for improvement.
4. Analyze change management policy for new functionalities.

Design

1. Design user interface (UX), system architecture and database schema with supervisor guidance.

2. Research and plan integration of various technologies (Microsoft connectors, On-Premises Gateway, Role-Based Access Control using Azure Active Directory, Power Automate and other technologies that may be required after research).

Development

- Setup development environment and integrate with required technologies.
- Develop and implement database integration and test connection between cloud app and on-premises database.
- Develop new system based on design specifications, implement required components between systems and gather feedback from supervisor and stakeholder at each milestone.
- Perform unit testing during development and troubleshoot issues found.
- Gather iterative feedback from stakeholder and supervisor.

Testing and Deployment

1. Develop and implement database integration and connection between cloud app and on-premises database.
2. Develop and carryout final test plan of the system before publishing to production.
3. Deploy to production environment and troubleshoot issues related to production environment.

Documentation and Training

1. Develop simple user guide and conduct a training session for ITS staff.

3.6 Deliverables Produced

Requirements and Analysis

- System requirements, current system workflow diagram and gap analysis interview documents.

Design

- User Interface (UX) mockup design, system architecture diagram and Database schema design documents.

Development

- Power app canvas app, Power Automate flows, SQL database and RBAC implementations explanation documentation.

Testing and Training

- Comprehensive test plan and simple user guide documentation.

Course Project Management

- Project proposal, weekly progress report, final project report, final project presentation documents.

4. Results and Outcomes

4.1 Evidence of Deliverables

To demonstrate the success in upgrading Lincoln University change management system, the final presentation will show case the following:

- A visual side-by-side comparison of old and new system, explanations highlighting improvements. Show casing of how each requirement has been met in the final system to stakeholder requirements.
- Unit testing results and a visual demonstration of application's workflow.

4.2 Testing/validation

To test the new change management system for Lincoln University, I plan to use the following testing strategy during development and final deployment.

Unit Testing:

- Perform unit testing of components and functionality of the Power Apps as developed.
- Manually verify data flow between different components of the system when developing integration.

Comprehensive Test Plan:

- Individually test different (RBAC) roles in application and their functionality.
- Perform Power Automate flows testing manually by using the application and triggering application workflows.
- Test the application of different web browsers to ensure compatibility.

- Test complete business workflow process.
- Test and verify all functionality works as intended in the production environment.

Feedback:

- Iterative feedback improvements from industry supervisor and stakeholder.

5. Milestones

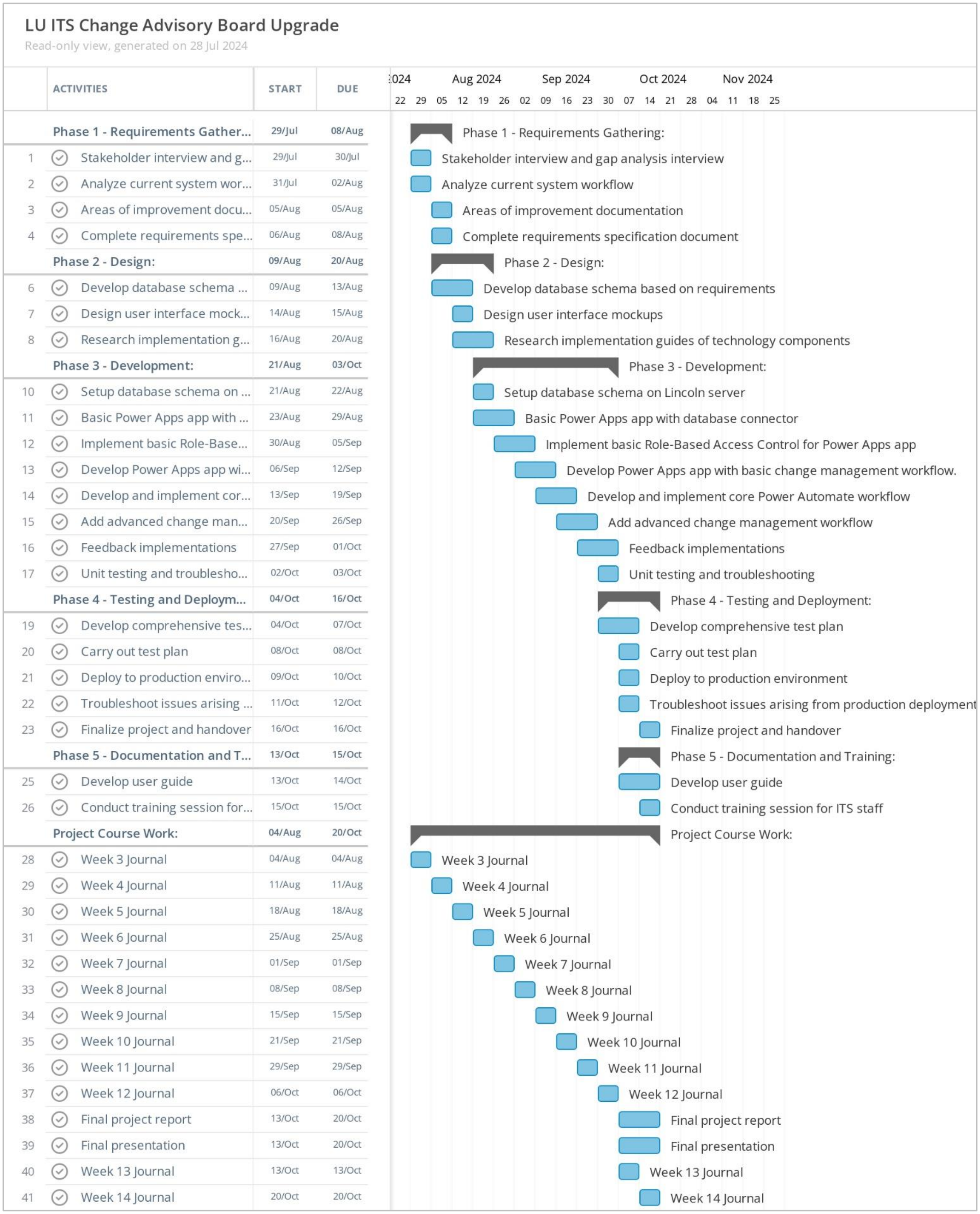
MILESTONES	DEADLINE
1. **Phase 1: Requirements Gathering** • Gap Analysis interview • System requirements document	8thAugust 2024
2. **Phase 2: Design** • Database schema, User Interface (UX), Change management workflow and Microsoft technology implementation research.	20th August 2024
3. **Phase 3: Development** • Core Power App development, Unit Testing with troubleshooting and gather give and implement feedback changes.	3rd October 2024
4. **Phase 4: Testing and Deployment** Deploy to production environment and conduct comprehensive test plan	16th October 2024
5. **Phase 5: Documentation and Training** • Develop user guide and conduct training session.	15th October 2024
6. **Project course task and reporting**	20th October 2024

6. Schedule of Activities – Timeline for completion

SCHEDULE OF ACTIVITIES	Time Allocation
1. **Requirements gathering and Gap Analysis interview** • Stakeholder interview and gap analysis interview • Analyze current system workflow • Areas of improvement documentation • Complete requirements specification document	2 days 3 days 1 days 4 days
2. **Design** • Develop database schema based on requirements • Design user interface mockups • Research implementation guides of technology components	3 days 2 days 3 days

3. **Development** • Setup database schema on Lincoln server and successful integration of on-premises SQL database data gateway to cloud app • Basic Power Apps app setup with database connector • Implement basic Role-Based Access Control for Power Apps app • Develop Power Apps app with basic change management workflow. • Develop and implement core Power Automate workflow • Add advanced change management workflow • Feedback implementation • Unit testing and troubleshooting	2 days 5 days 5 days 5 days 5 days 5 days 3 days 2 days
4. **Testing and Deployment** • Develop comprehensive test plan • Carry out test plan • Deploy to production environment • Troubleshoot arises from production deployment	2 days 1 day 2 days
5. **Documentation and Training** • Develop user guide • Conduct training session for ITS staff • Finalize project and handover	2 days 1 day 1 day
6. **Project course task and reporting** • Weely Journals • Final project report • Final presentation	Weekly 5 days 5 days

6.1 Gantt Chart of Activities



LU ITS Change Advisory Board Upgrade

Read-only view, generated on 28 Jul 2024

	ACTIVITIES	START	DUE
	Phase 1 - Requirements Gather...	**29/Jul**	**08/Aug**
1	✓ Stakeholder interview and g...	29/Jul	30/Jul
2	✓ Analyze current system wor...	31/Jul	02/Aug
3	✓ Areas of improvement docu...	05/Aug	05/Aug
4	✓ Complete requirements spe...	06/Aug	08/Aug
	Phase 2 - Design:	**09/Aug**	**20/Aug**
6	✓ Develop database schema ...	09/Aug	13/Aug
7	✓ Design user interface mock...	14/Aug	15/Aug
8	✓ Research implementation g...	16/Aug	20/Aug
	Phase 3 - Development:	**21/Aug**	**03/Oct**
10	✓ Setup database schema on ...	21/Aug	22/Aug
11	✓ Basic Power Apps app with ...	23/Aug	29/Aug
12	✓ Implement basic Role-Base...	30/Aug	05/Sep
13	✓ Develop Power Apps app wi...	06/Sep	12/Sep
14	✓ Develop and implement cor...	13/Sep	19/Sep
15	✓ Add advanced change man...	20/Sep	26/Sep
16	✓ Feedback implementations	27/Sep	01/Oct
17	✓ Unit testing and troublesho...	02/Oct	03/Oct
	Phase 4 - Testing and Deploym...	**04/Oct**	**16/Oct**
19	✓ Develop comprehensive tes...	04/Oct	07/Oct
20	✓ Carry out test plan	08/Oct	08/Oct
21	✓ Deploy to production enviro...	09/Oct	10/Oct
22	✓ Troubleshoot issues arising ...	11/Oct	12/Oct
23	✓ Finalize project and handover	16/Oct	16/Oct
	Phase 5 - Documentation and T...	**13/Oct**	**15/Oct**
25	✓ Develop user guide	13/Oct	14/Oct
26	✓ Conduct training session for...	15/Oct	15/Oct
	Project Course Work:	**04/Aug**	**20/Oct**
28	✓ Week 3 Journal	04/Aug	04/Aug
29	✓ Week 4 Journal	11/Aug	11/Aug
30	✓ Week 5 Journal	18/Aug	18/Aug
31	✓ Week 6 Journal	25/Aug	25/Aug
32	✓ Week 7 Journal	01/Sep	01/Sep
33	✓ Week 8 Journal	08/Sep	08/Sep
34	✓ Week 9 Journal	15/Sep	15/Sep
35	✓ Week 10 Journal	21/Sep	21/Sep
36	✓ Week 11 Journal	29/Sep	29/Sep
37	✓ Week 12 Journal	06/Oct	06/Oct
38	✓ Final project report	13/Oct	20/Oct
39	✓ Final presentation	13/Oct	20/Oct
40	✓ Week 13 Journal	13/Oct	13/Oct
41	✓ Week 14 Journal	20/Oct	20/Oct

:024
Aug 2024
Sep 2024
Oct 2024
Nov 2024
22 29 05 12 19 26 02 09 16 23 30 07 14 21 28 04 11 18 25
Phase 1 - Requirements Gathering:
Stakeholder interview and gap analysis interview
Analyze current system workflow
Areas of improvement documentation
Complete requirements specification document
Phase 2 - Design:
Develop database schema based on requirements
Design user interface mockups
Research implementation guides of technology components
Phase 3 - Development:
Setup database schema on Lincoln server
Basic Power Apps app with database connector
Implement basic Role-Based Access Control for Power Apps app
Develop Power Apps app with basic change management workflow.
Develop and implement core Power Automate workflow
Add advanced change management workflow
Feedback implementations
Unit testing and troubleshooting
Phase 4 - Testing and Deployment:
Develop comprehensive test plan
Carry out test plan
Deploy to production environment
Troubleshoot issues arising from production deployment
Finalize project and handover
Phase 5 - Documentation and Training:
Develop user guide
Conduct training session for ITS staff
Project Course Work:
Week 3 Journal
Week 4 Journal
Week 5 Journal
Week 6 Journal
Week 7 Journal
Week 8 Journal
Week 9 Journal
Week 10 Journal
Week 11 Journal
Week 12 Journal
Final project report
Final presentation
Week 13 Journal
Week 14 Journal