

**ICT200 – Task 2**

**Author**: Torin Flanagan

**ID:** 1169130

**Email:** TDF008@student.usc.edu.au

**Date:** 22nd May 2024

***Podunk Council Library Report***

**Table of Contents**

[Executive Summary 2](#_Toc161390616)

[Introduction 3](#_Toc161390617)

[1. Identification of System Requirements 4](#_Toc161390618)

[2. Critical Analysis of User Stories 5](#_Toc161390619)

[3. Use Case Descriptions 8](#_Toc161390620)

[4. Use Case Diagram 10](#_Toc161390621)

[5. Domain Model Class Diagram 11](#_Toc161390622)

[6. Design Class Diagram 12](#_Toc161390623)

[7. References 14](#_Toc161390624)

Executive Summary

The Podunk Council Library was undergoing a comprehensive digital transformation project to improve operational efficiency and community service delivery. The previous outdated systems for volunteer activities and facility bookings were developing operational inefficiencies and limited-service offerings, this then produced negative user experience and increased administrative overhead. Due to the library’s previous lack to utilise digital technologies along with the inability to expand event offerings, community engagement continued to decrease. Additionally, without the use of a centralised platform to provide training materials to volunteers, efficiency in training and safety compliance was being negatively impeded.

Resources including the System Vision Document, user stories, and the Stakeholder Analysis Matrix were used to address the previously mentioned challenges. The system Vision Document outlined the system capabilities and future benefits the Podunk Council Library were visioning. Meanwhile, user stories offered insight into each user’s requirements. Lastly, the Stakeholder Analysis Matrix provided further details for key engagement strategies and priorities.

Functional requirements addressed for the Podunk Council Library were the Queensland Government’s user management system, support for event and program management, the digital volunteer coordination platform, the online booking system for facilities, and the staff scheduling management tools. Non-functional requirements that were managed for the Podunk Council Library included reliability, scalability, usability, security, integration, performance, and compliance.

The newly proposed system aimed for a modernised authentication process, streamlined event management, optimised volunteer coordination, enhanced facility access, and improved staff scheduling. Additionally, it ensured system reliability, scalability, usability, security, integration, performance, and compliance. This holistic approach positioned the library to better serve the community as a dynamic hub for knowledge and engagement.

Introduction

The Podunk Council Library faces critical challenges, prompting a comprehensive digital transformation project to enhance operational efficiency and community service delivery. Currently reliant on outdated systems for volunteer activities and facility bookings, the library experiences operational inefficiencies and limited-service offerings. These outdated processes lead to increased administrative overhead, errors, and negative user experiences, furthermore, the library’s inability to expand its event range and utilise digital technologies hampers community engagement. The lack of a centralised platform for training materials poses another obstacle, hindering volunteer training and safety compliance. Addressing these bottlenecks is crucial for modernising services and meeting user expectations and efficiency standards.

To address these needs, resources including the System Vision Document (Appendix A), User Stories (Appendix B), and the Stakeholder Analysis Matrix (Appendix C) have been provided. The System Vision Document aids in understanding the problem, system capabilities, and future benefits. User stories offer insights into user requirements for integration. The Stakeholder Analysis Matrix provides details on stakeholder engagement strategies, priorities, and contributions.

1. Identification of System Requirements

**Functional requirements:**

1. Integration with Queensland Government’s user management: Seamless integration with government system for authentication and management.
2. Event and Program Management: Support for scheduling, ticketing, registrations, reminders, and feedback, surpassing limitations of Eventbrite and Excel.
3. Volunteer Coordination Platform: Digital platform for volunteer management, replacing the Excel-based system.
4. Online Booking:Inclusion of online booking system for facility reservations, with real-time availability checks.
5. Staff Scheduling and Management:Efficient scheduling, task assignments, performance tracking, and communication tools.

**Non-Functional requirements:**

1. Reliability:24/7 availability with minimal downtime for data integrity.
2. Scalability**:** Ability to handle increased volumes without performance degradation.
3. Usability:Intuitive interface for both skilled and unskilled users.
4. Security:Secure data storage and transmission with access tools.
5. Integration: Seamless integration with existing systems and provision of APIs.
6. Performance: Responsive system with fast load times and efficient user interaction.
7. Compliance:Adherence to data protection regulations and accessibility standards.

**How will these requirements address the challenges outlined in the problem statement?**

The proposed system promises to address operational challenges and enhance community engagement at the Podunk Council Library. Integration with the Queensland Government’s user management will modernise authentication, reducing reliance on outdated ticketing systems. Enhancements in event and program management will streamline processes, allowing the library to broaden offerings and encourage greater participation. Transitioning to a digital volunteer platform will improve engagement and efficiency, optimising volunteer contributions and safety adherence. Introduction of an online booking system will enhance facility access, improving user satisfaction and operational efficiency. Implementing staff scheduling systems will foster a positive work environment, boosting productivity and staff retention. Non-functional requirements ensure system reliability, scalability, usability, security, integration, performance, and compliance, positioning the library for sustainable success, serving the community as a dynamic hub for knowledge and engagement.

2. Critical Analysis of User Stories

**Critically analyse the user stories to identify any gaps, ambiguities, or inconsistencies, and list five (5) that you believe require further requirements gathering activities.**

Upon reviewing the provided user stories (APPENDIX B) and the stakeholder analysis matrix (APPENDIX C), several areas warrant further requirements gathering activities to ensure comprehensive coverage and alignment with the stakeholder needs:

1. User Story 1, Check Out Books Online: While the user story addresses the need for an online book checkout, it lacks specification regarding the process, such as the steps involved, supported formats (e.g. e-books), and potential constraints (e.g. loan periods). Further clarification is needed to ensure all aspects of the online checkout are appropriately covered.
2. User Story 7, Schedule Library Events Online: While this user story outlines the need for scheduling library events online, it does not specify the features or functionalities required for effective event management. Additional information is needed on event creation, scheduling options, attendee registration, and event promotion to ensure the system meets the needs of both event organisers and attendees.
3. User Story 23, Manage Volunteer Schedules Online: This user story addresses the need for managing volunteer schedules online but lacks details on specific requirements, such as scheduling tools, communication features, and volunteer task assignments. Further exploration is needed to understand the complexities of volunteer management and ensure the system adequately supports these activities.
4. User Story 28, Oversee Booking System for Library Spaces: While this user story addresses the oversight of the booking system for library spaces, it does not specify the key functionalities required for effective space management. Additional requirements gathering is needed to identify features such as real-time availability, booking restrictions, and space utilisation analytics to ensure efficient space management.
5. User Story 31, Set Up Parental Controls on Library Account: While this user story addresses the need for parental controls, it lacks details on the specific controls and restrictions required. Further exploration is needed to identify the desired parental control features, such as content filtering, access restrictions, and monitoring capabilities, to ensure the system meets the needs of parents and guardians.

In summary, these user stories require further requirements gathering activities to clarify specific functionalities, features, and constraints to ensure the resulting system adequately addresses stakeholder needs and expectations.

**For each of the five (5) listed user stories, explain how you would recommend addressing the issues with them, including specific questions you might ask specific stakeholders.**

* User Story 1, Check Out Books Online:
* Work with library management to understand online checkout process. Queries include:
  + Current physical checkout steps; adaptions for online platform?
  + Loan periods, renewals, hold constraints?
  + Supported material formats for online checkout (e.g. books, e-books, audiobooks)?
  + System handling of reservations and availability for popular items?
* User Story 7, Schedule Library Events Online:
  + Collaborate with program coordinators and librarians for event scheduling. Queries include:
    - Event types and common scheduling scenarios?
    - Event creation, scheduling, and management in the system?
    - Event listing details (e.g. descriptions, locations, dates)?
    - Attendee registration, ticketing, event promotion specifics?
    - Handling recurring events or cancelations?
* User Story 23, Manage Volunteer Schedules Online:
  + Consult volunteer coordinators and staff for schedule management. Queries include:
    - Current scheduling processes, pain points?
    - Required tools or features for task assignment and volunteer communication?
    - Volunteer access to schedules, assignment notifications?
    - Tracking volunteer hours, evaluating performance specifics?
* User Story 28, Library Space Booking System Oversight:
  + Collaborate with facilities managers for space booking. Queries include:
    - Library spaces to be booked, typical scenarios?
    - Real-time availability display, booking requests?
    - Booking restrictions (e.g. capacity, recurring bookings)?
    - Reporting, analytics for space utilisation, booking trends?
* User Story 31, Parental Controls Setup:
  + Understand parental requirements for controls. Queries include:
    - Restricted content or activities for children’s accounts?
    - Age-based restrictions, filtering criteria?
    - Parent setup, management of controls in the system?
    - Privacy, security concerns regarding controls?

3. Use Case Descriptions

Use case descriptions are a text-based narrative of step-by-step interactions explaining the functionality between the actor and the system, it describes the outcomes of an action taken to accomplish a goal within a system (Sindre and Opdahl, 2001). The below use case descriptions are based on managing volunteer schedules online and the other is about overseeing bookings for library spaces, both use case will include a user story, actors, pre-conditions, workflow steps, and post-conditions.

**Use Case Description 1:**

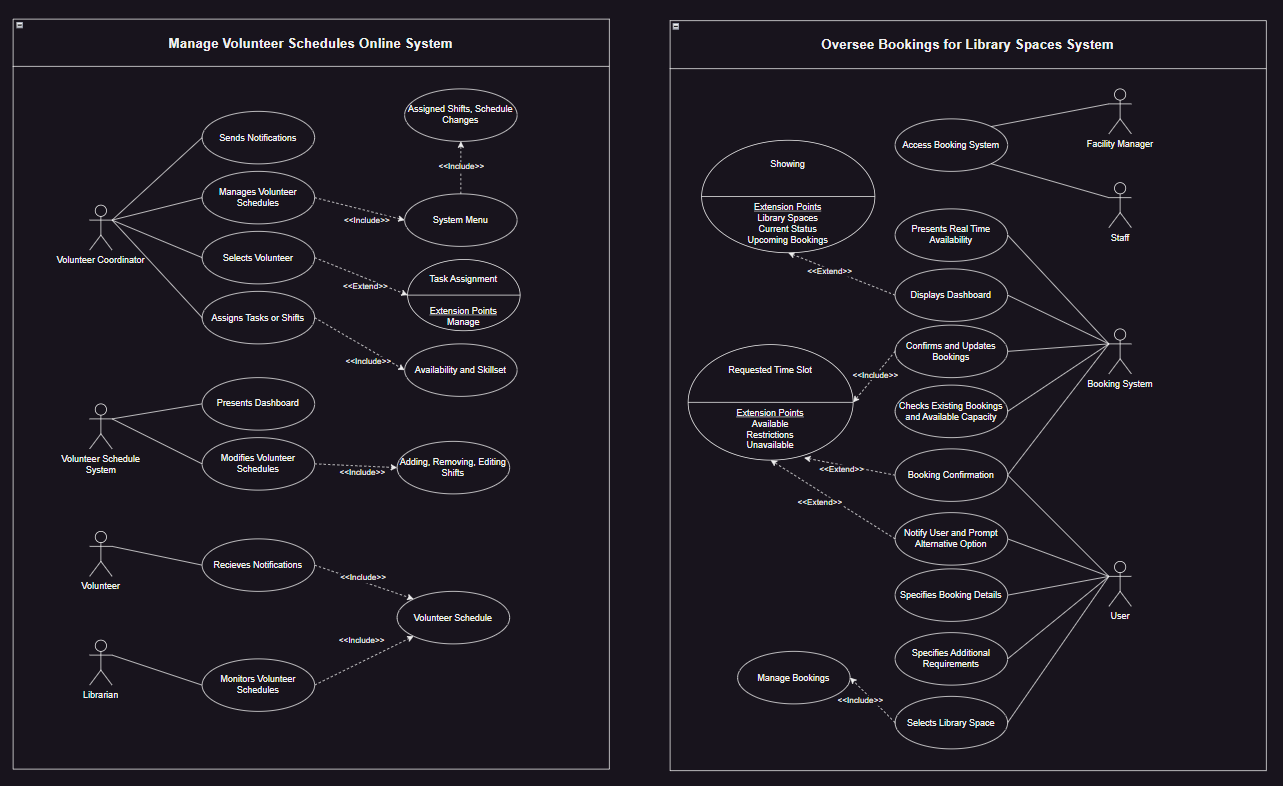
|  |  |
| --- | --- |
| **User Story** | Be able to manage volunteer schedules and assignments online to efficiently organise volunteer efforts. |
| **Use Case Name** | Manage Volunteer Schedules Online. |
| **Actors** | * Volunteer coordinator (primary actor). * Volunteer (secondary actor). * Librarian (secondary actor). |
| **Pre-conditions** | * System is operational and accessible to authorised users. * Volunteer coordinator or Librarian have logged into the system with appropriate permissions. * Volunteer profiles and schedule templates are present in the system. |
| **Workflow Steps** | 1. Volunteer Coordinator selects the ‘Manage Volunteer Schedules’ option from the system menu. 2. System presents a dashboard displaying available volunteers, assigned tasks, and upcoming shifts. 3. Volunteer coordinator selects a specific volunteer or task assignment to manage. 4. System allows for modification of volunteer schedules, including adding, removing, or editing shifts. 5. Volunteer Coordinator assigns specific tasks or shifts to volunteers based on availability and skillset. 6. Volunteer Coordinator sends notifications to volunteers regarding assigned shifts and schedule changes. 7. Volunteers receive notifications through the system or email and confirm availability for assigned shifts. 8. Librarian monitors volunteer schedules and communicates any updates or changes. |
| **Post-conditions** | * Volunteer schedules are updated in real-time through the system, reflecting assigned tasks and shifts. * Volunteers receive timely notifications about assigned shifts, improving communication and coordination. * Librarians can efficiently manage volunteer resources, ensuring coverage for library activities and events. * Volunteer Coordinator can track volunteer hours, assignments, and performance for reporting and evaluation purposes. |

**Use Case Description 2:**

|  |  |
| --- | --- |
| **User Story** | Supervise the booking system to optimise space utilisation and ensure proper maintenance. |
| **Use Case Name** | Oversee Bookings for Library Spaces. |
| **Actors** | * Facility Manager (Primary Actor). * Staff Responsible for Space Management (Secondary Actor). * System Administrator (Secondary Actor). |
| **Pre-conditions** | * System is operational and accessible for authorised users. * Relevant library spaces are defined and available for bookings. |
| **Workflow Steps** | 1. Facilities manager or staff responsible for space management access the booking system. 2. System displays a dashboard showing available library spaces, current status, and upcoming bookings. 3. User selects a specific library space to manage bookings. 4. System presents options for real time availability. 5. User initiates booking request by specifying the desired date, time, and duration of the booking. 6. If applicable, user specifies additional requirements or equipment needed for the booking. 7. System checks for conflicts with existing bookings and available capacity for the requested time slot. 8. If requested time slot is available and within booking restrictions, the system confirms the booking and updates the booking schedule accordingly. 9. If requested time slot is not available or does not meet booking restrictions, the system notifies the user and prompts for alternative options. 10. After confirming the booking, the system generates a booking confirmation for the user and updates the booking schedule for the selected library space. |
| **Post-conditions** | * Booking system reflects updated booking schedule for the selected library space. * The user receives a confirmation of the successful booking. |

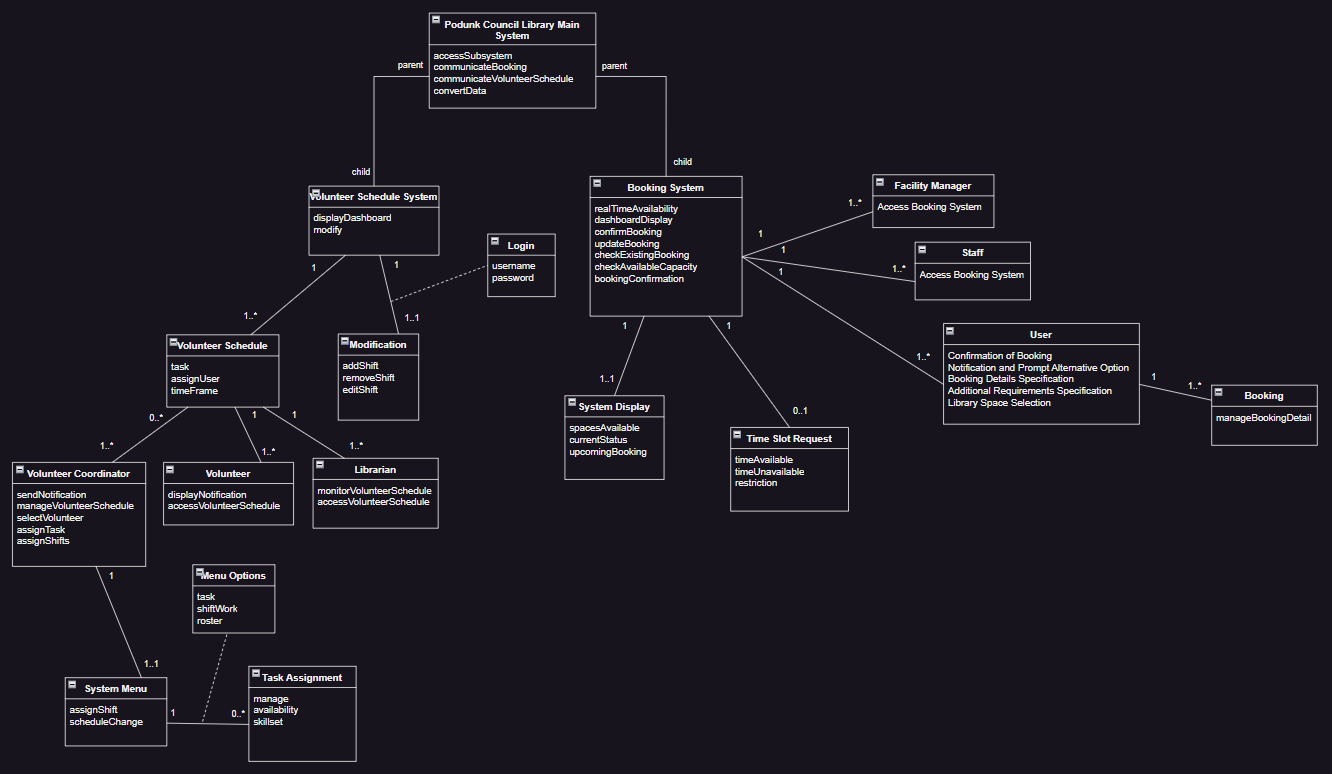
4. Use Case Diagram

Use case diagrams serve as a means to identify the specified requirements for a software system, they function as a communication tool facilitating an understanding between software system requirements developers and potential users regarding the essential needs of a software system (Shen & Liu, 2003). The following use case diagrams illustrate the connections between the necessary actors and the actions they are associated with and execute.

****Figure 1: Use Case Diagram**

5. Domain Model Class Diagram

Domain model class diagrams incorporate details that encompass various elements represented in UML, these include all identified methods in collaboration diagrams, navigation within association ends, the scope and type of attributes, and newly discovered associations during the design phase (Brahim, Omar, & Taoufiq, 2013). It is important to note that a requirement for the below domain model class diagram is to not include any methods due to the diagram’s purpose being to communicate the logical structure of the sub-systems, their key entities, and associations between them.

**Figure 2: Domain Model Class Diagram**

6. Design Class Diagram

Design class diagrams show the static structure of the system being modelled, the diagram specifically shows the entities within the system, along with each entity’s internal structure and relationships with other entities inside the system (Bell, 2003). The provided design class diagrams depict the online space booking system and the volunteer schedule system implemented for the Podunk Council Library, these two subsystems are linked as child associations to the main system of the Podunk Council Library, which serves as the parent system.

**Figure 3: Design Class Diagram Booking System Interface***A screenshot of a computer flowchart

Description automatically generated*

**A screenshot of a computer program

Description automatically generatedFigure 4: Design Class Diagram Volunteer Schedule System**

7. References

* Bell, D. (2003). UML Basics Part III: The Class Diagram. In *The Rational Edge* (p. 13). https://tildesites.bowdoin.edu/~allen/courses/cs260/readings/UMLclassdiagrams.pdf
* Brahim, B., Omar, E. B., & Taoufiq, G. (2013). Generating operations specification from domain class diagram using transition state diagram. *International Journal of Computer and Information Technology*, *2*(1), 7. academia.edu. https://d1wqtxts1xzle7.cloudfront.net/68183778/Generating\_operations\_specification\_from20210718-3526-1w4f05q.pdf?1626656016=&response-content
* Satzinger, J.W., Jackson, R.B. and Burd, S.D. (2015). *Systems Analysis and Design in a Changing World*. [online] 20 Channel Center Street Boston, MA 02210 USA: Cengage Learning, p.512. Available at: https://ebookcentral.proquest.com/lib/usc/detail.action?docID=4771015 [Accessed 22 May 2024].
* Shen, W., & Liu, S. (2003). Formalization, Testing and Execution of a Use Case Diagram. *FormalMethods and Software Engineering*, 68–85.https://doi.org/10.1007/978-3-540-39893-6\_6
* Sindre, G., & Opdahl, A. (2001, June). *Templates for Misuse Case Description*. Research Gate. https://www.researchgate.net/profile/Andreas-Opdahl/publication/2873777\_Templates\_for\_Misuse\_Case\_Description/links/09e4150cbe4e9a3eee000000/Templates-for-Misuse-Case-Description.pdf