2024

# Project Completion Report.

ST10083252

AVARN SEWLAL

PROG7312

# Project Completion Report: Municipal Services WPF Application

## Project Overview

The **Municipal Services WPF Application** is a feature-rich desktop platform designed to streamline municipal issue reporting, local event management, and service request status tracking. The application empowers users to report and track municipal issues, browse and receive recommendations for local events, and manage service requests while visualizing their dependencies.

The project aims to enhance community engagement and improve municipal service efficiency through intuitive features such as:

1. **Issue Reporting**
   - A streamlined process for reporting municipal issues, complete with media attachment capabilities.
2. **Local Events Management**
   - Event browsing, personalized recommendations, and real-time notifications to keep users informed about community events.
3. **Service Request Status Management**
   - Tracking service requests and managing their dependencies using efficient data structures.

This report details the challenges faced, solutions implemented, and key learnings from the completion of all three major components of the project.

---

## Challenges Faced and Solutions

### 1. Issue Reporting

### Challenges

1. **Form Validation and Media Attachments**: Ensuring robust form validation while allowing users to upload images in various formats (e.g., .jpg, .jpeg, .png).

2. **User Feedback**: Providing clear and immediate confirmation to users after a successful submission.
3. **Scalability**: Designing a system capable of handling a growing number of issue reports.

**Solutions**

1. **Comprehensive Validation**: Implemented validation logic to ensure all required fields were completed, and uploaded files adhered to supported formats.
2. **Reset and Confirmation**: Added a form reset function with confirmation messages to streamline repeated submissions.
3. **Efficient Data Handling**: Designed the backend to efficiently store and process issue data, preparing it for future scalability.

---

### 2. Local Events Management

**Challenges**

1. **Dynamic Filtering and Sorting**: Managing a dynamic list of events that users could filter by name, category, or date while ensuring the next upcoming event was highlighted.
2. **Personalized Recommendations**: Developing a system that analysed user search history to provide relevant event suggestions.
3. **Notification System**: Implementing a notification system to alert users about events occurring in the next 7 days.

**Solutions**

1. **Optimized Filtering and Sorting**: Used LINQ queries and WPF data binding to dynamically update and display events based on user preferences.
2. **Recommendation Engine**: Built a lightweight recommendation system that analysed user search history to suggest relevant events.

3. **Asynchronous Notifications**: Implemented a queue-based system that processed and triggered notifications without affecting the app's performance.

---

**3. Service Request Status Management**

**Challenges**

1. **Efficient Storage and Retrieval**: Storing and retrieving service requests efficiently as the dataset grew.
2. **Dependency Visualization**: Representing dependencies between service requests and preventing inconsistencies (e.g., cyclic dependencies).
3. **Real-Time Updates**: Synchronizing and updating multiple data structures (BST and Graph) in real-time.

**Solutions**

1. **Binary Search Tree (BST)**: Used a BST to store service requests for efficient search, insertion, and retrieval operations.
2. **Graph Representation**: Represented dependencies using a Graph with an adjacency list, enabling quick updates and visualizations.
3. **Cycle Detection**: Implemented depth-first search (DFS) to detect and prevent cyclic dependencies.

---

**Key Learnings**

**Skills Acquired**

1. **Data Structures Mastery**
   - Gained hands-on experience with BSTs and Graphs for managing hierarchical and relational data, respectively.
2. **Asynchronous Programming**

    - Learned to handle long-running operations in the background, ensuring a responsive user interface.

3. **Event-Driven Development**
    - Developed robust event-driven systems for features like notifications and real-time updates.
4. **User-Centric Design**
    - Improved proficiency in creating intuitive and accessible WPF interfaces for diverse functionalities.

**Problem-Solving Approaches**

1. **Incremental Testing**
    - Developed and tested individual modules separately before integrating them into the application.
2. **Error Handling**
    - Built robust error-handling mechanisms to manage edge cases gracefully.
3. **Optimization**
    - Profiled and optimized performance-heavy operations, such as Graph traversals and in-order traversal of the BST.

---

**Delivered Features**

**1. Issue Reporting**

- Users can report municipal issues with details like location, category, and description.
- Supports media file attachments for added context.
- Confirmation messages provide feedback on successful submissions.

**2. Local Events Management**

- Users can browse and filter community events by category, date, or name.
- Personalized recommendations based on search history enhance user engagement.
- Notifications alert users about events happening within the next 7 days.
- Recently viewed events are tracked for quick access.

**3. Service Request Status Management**

- Users can add, view, and search service requests using efficient BST storage.
- Dependencies between requests are represented and visualized using a Graph.
- The application prevents invalid dependencies through cycle detection.

---

**Future Recommendations**

1. **Enhanced Visualizations**
   - Add graphical representations for the Graph and BST to improve data visualization.
2. **Mobile Integration**
   - Extend the application to mobile platforms to reach a wider audience.
3. **Database Integration**

- Integration will allow for Save issue reports and event data to be real time inputted and long term stored without the need for it to be hardcoded

4. **User Account Integration**
   - Allow users to create accounts to track their submitted issues, viewed events, and service requests.

## Conclusion

The **Municipal Services WPF Application** successfully integrates issue reporting, local events management, and service request tracking into a single cohesive platform. By overcoming challenges in data handling, dependency management, and user experience design, the project delivers a robust, user-friendly application that addresses real-world municipal needs.

The project provided significant learning opportunities, especially in data structure implementation, WPF development, and system optimization. It sets a strong foundation for future enhancements and broader implementation across other platforms.