

PROG7312 st10091991

Project Completion Report



November 18, 2024

vcwaterfall

midrand

**Introduction**This report highlights the completion of a municipality application developed using .NET MAUI. Implementing features for issue reporting, service request, local event and functions announcements are the main objective. Several sophisticated programming ideas, such as data structures, UI/UX design, and backend integration, were used during the development process to provide an application that is easy to use and satisfies the desired specifications.  
  
**Challenges Faced in Part 3**

Implementing the Service Request Status feature, which allowed users to report problems, monitor the progress of service requests, and move between different status phases, was implemented in the third phase of the project. The work included automating status updates, integrating intricate data structures, and making sure the application operated seamlessly with an intuitive user interface. There was however 5 main challenges that were faced :  
  
**Challenge 1** : Service request automation  
**Problem :** Due to the very little information that was provided in part 3 on how the requested service status would be taken to the next phase of the problem status hierarchy made implementation of this feature very odd, imagine having to build a house door into a house however there is no door frame from where the door can be placed upon, the same concept applies to the service requests where there was not really a definitive way in which the status would be updated either from an admin user or an employee of the municipality itself.  
  
**Solution :** Implementing a background process or service that routinely verifies and updates the statuses of all pending service requests locally from the binary tree was the answer. I was able to automate the procedure by using a time-based trigger (via timed events or a background task). An effective data retrieval technique and the inclusion of a Status field in every request guaranteed that the status updates were implemented appropriately. This solution is locally implemented only and not by a database meaning that there is no online or local database connection to the application.  
  
**Challenge 2**: Handling Linked Issues with Service Requests  
**Problem**: Ensuring that problems customers reported could be accurately associated with new service requests was another difficulty. Depending on the situation, when users report a problem, the application must either link the issue to an existing service request or create a new one. This logic necessitated maintaining several data relationships and the interactions between the application's numerous components.

**Solution**: I employed sophisticated data structures like trees and graphs to solve this problem. These structures made it easier to efficiently arrange and retrieve service request data. To represent the several stages of the request, the ServiceRequest class was created to receive an IssueReport object and a status number. A seamless flow was made possible by the hybrid approach, which made it simple for consumers to connect a reported issue to a new service request.  
  
**Challenge 3**: UI and User Interaction

**Problem**: Another issue was making sure the user interface (UI) was not just useful but also easy to use and aesthetically pleasing. It was my responsibility to design an interesting user interface that seamlessly transitioned between pages and represented the status of each service call. The user experience felt a little static because the original design lacked fluid movements and transitions.

**Solution**: To get around this problem, I used animations and seamless button and screen transitions. For instance, page transitions employed slide and fade animations, and buttons exhibited bounce effects when clicked. This improved the user experience by making the interface more dynamic and increasing the app's engagement.

**Challenge 4**: Error Handling and Debugging  
**Problem**: I ran across several issues with threading, database queries, and user interface interactions throughout the development stage. The app's inability to get to the Landing Screen when attempting to register a new user with biometric identification enabled was one of the most annoying problems.

**Solution**: To prevent ANR (Application Not Responding) problems, the answer was to move processing pertaining to biometrics off the main thread. To make sure that the application could continue without crashing even if biometric registration failed, I also included a fallback mechanism.  
  
**Key Learnings**  
During the study, several important insights were discovered. My comprehension of application development has improved because of these lessons, which have also improved my programming and problem-solving abilities.

1. Advanced Data Structures and Algorithms:  
   I developed a thorough understanding of heaps, graphs, and trees and discovered how to use them to store and handle user data and service requests. The success of the project depended heavily on the ability to efficiently modify and analyse enormous databases.
2. UI/UX Design:  
   It became clear how important seamless user interactions are. I discovered how to utilise XAML to make responsive layouts and how to employ animations to improve user experience. Comprehending the harmony between practicality and beauty has become an essential component of my development methodology.
3. Problem-Solving Approaches:  
   I took a more methodical approach to problem-solving during the project, concentrating on decomposing difficult problems into smaller, more doable ones. This assisted me in overcoming obstacles like making sure UI transitions were seamless and automating status updates.

**Conclusion**

This endeavour has been a worthwhile educational endeavour. Along with improving my technical abilities, I also gained a better grasp of how to tackle intricate system designs and user interface implementations. Advanced programming techniques and problem-solving strategies were used to successfully overcome the difficulties encountered, especially regarding automating service request status updates and managing data linkages. The product is a useful, interactive, and intuitive application.  
  
I am eager to keep honing my abilities and using them in next projects, especially ones that call for incorporating more sophisticated backend features and further honing user interfaces.