# **We Got You - Municipal Services Application Report**

## **Overview**

The We Got You application, which aims to improve community involvement and quicken municipal service management in South Africa, was successfully completed, as this report details. The app gives users an easy way to view service request statuses, identify any problems, provide feedback, and keep up with announcements and events in their community. This project's main goals were to enhance user experience, promote transparency between local government and citizens, and implement useful features using advanced data structures.

## **Implemented Components and Features**

### **1. Main Menu**

The Main Menu serves as the application's central hub, providing access to key features:

* **Report Issues**: Users can submit reports, specify locations, provide details, and attach media files.
* **Local Events and Announcements**: Displays upcoming events with search, filtering, and sorting options.
* **Status Request History:** Displays the status of a request, whether it has been completed, pending, etc. And who it has been updated by.
* **Show Recommendations**: Suggests relevant events based on user search history.
* Feedback system: form in which a user can provide feedback on any services provided by the municipality.

### **2. Reporting Issues**

* Users can report municipal issues such as infrastructure damage and environmental concerns.
* Required fields include **location, category, priority, and a detailed description**.
* Users can attach media files to provide additional context.
* Each submission generates a unique **RequestId** for tracking purposes.
* The form includes **validation** to ensure required fields are filled before submission.
* Users can navigate back to the main menu easily.
* To encourage engagement, **thank-you messages** are displayed after feedback submission as well as a loading bar to let them know how much of the form is complete.

### **3. Service Request Status**

* Users can check the status of their reports by entering their **RequestId**.
* The system provides details such as **status, submission date, and updates**.
* This feature promotes **transparency**, allowing citizens to track their submissions.

### **4. Feedback Collection System**

* Users can **rate** municipal services and provide optional comments.
* Feedback helps municipal authorities assess user satisfaction and identify areas for improvement.

### **5. Local Events and Announcements**

* Displays **upcoming events and community announcements**.
* Users can **search and filter** events by category and date.
* A dedicated **FilterWindow** was created to enhance sorting capabilities.
* **Smart Recommendations** were implemented, analyzing user search history to suggest relevant events.

## **Utilized Data Structures**

1. **List**
   1. Stores reported issues and service requests sequentially (C# Corner, 2023).
   2. Manages service requests for easy navigation and manipulation (C# Corner, 2023).
2. **Dictionary**
   1. Enables quick lookup and management of service requests using **unique RequestId keys (Geeksforgeeks, 2024)**.
   2. Ensures constant-time access for **adding, updating, and retrieving** data (Geeksforgeeks, 2024).
3. **Queue**
   1. Manages service requests in a **first-come, first-served** manner.
   2. Ensures efficient processing of pending requests.
4. **SortedDictionary**
   1. Keeps service requests **organized** based on timestamps or priority levels (Geeksforgeeks, 2024).
   2. Facilitates efficient processing in order of importance (Geeksforgeeks, 2024).
5. **Personalized Recommendation Algorithms**
   1. Analyzes user behavior and search history to suggest relevant events.
   2. Utilizes **sets and advanced logic** to track user preferences.
6. **Min Heap & Max Heap**:
   1. **Min Heap** prioritizes tasks with the lowest urgency for gradual processing (Geeksforgeeks, 2024).
   2. **Max Heap** ensures that the most critical issues are resolved first (Geeksforgeeks, 2024).
   3. Used for **prioritizing service requests and sorting urgent reports** efficiently (Geeksforgeeks, 2024).
7. **Basic Tree**:
   1. Stores event categories in a hierarchical structure (Geeksforgeeks, 2025).
   2. Allows efficient **category-based searching and filtering (Geeksforgeeks, 2025).**
8. **TreeView**:
   1. Displays **nested categories** in an intuitive format for better event browsing (C# Corner, 2023).
   2. Improves **navigation and user experience** when exploring events (C# Corner, 2023).
   3. To visualize the history of request statuses, I first attempted a graph-based method. A TreeView was used in its place because of its complexity and display limitations. Users can systematically expand and collapse their status history with this solution (C# Corner, 2023).

## **User Interface Design**

* **Theme**: Dark-themed interface with **white text** for optimal visibility.
* **Navigation**: Clear and consistent button placement for **intuitive usability**.
* **Layout Adjustments**: Optimized search bars, back buttons, and filter buttons for a seamless experience.
* **Dedicated Windows**: Implemented specific windows for event filtering and sorting to enhance usability.

## **Project Challenges and Solutions**

1. **Real-Time Data Updates**
   1. **Challenge**: Implementing a system that updates data efficiently.
   2. **Solution**: Used appropriate **data structures** to ensure quick and effective updates and retrieval.
2. **User-Friendly Interface**
   1. **Challenge**: Designing an interface that is both **intuitive and accessible**.
   2. **Solution**: Conducted **UI/UX testing** and adjusted controls and layout based on feedback. In the 3rd task, I decided to use a side-navigation which was much better in terms of spacing and overall layout.
3. **Handling Large Data Sets**
   1. **Challenge**: Managing large sets of reports and event data efficiently.
   2. **Solution**: Used **optimized data structures** like dictionaries and sorted collections. This also helped me to even store hardcoded data.
4. **Smart Recommendations**
   1. **Challenge**: Implementing an effective recommendation system which was poorly implemented in task 2.
   2. **Solution**: I managed to develop a robust **search-based recommendation** algorithm to analyze user preferences.
5. **Status History Visualization**

**Challenge:** I initially attempted to **use a graph** to display **status history**.   
**Issue:** The **graph-based approach** was too complex and difficult to integrate into the UI.   
**Solution:** In conclusion, I implemented a **TreeView**, which allows users to **expand/collapse** different stages of request status updates.

## **Quality Assurance and Testing**

* **Functional Testing**: Ensured all features worked as expected and produced correct outputs.
* **UI/UX Testing**: Assessed usability, accessibility, and visual consistency.
* **Performance Testing**: Evaluated the application’s responsiveness and data processing efficiency.

## **Technological Recommendations**

To further enhance the **We Got You** application, the following technological improvements are recommended:

1. **Cloud-Based Data Storage**
   1. **Justification**: I think Implementing **cloud storage solutions** (e.g., Firebase, Azure, or AWS) would enhance **data security, scalability, and real-time updates**.
2. **Push Notifications**
   1. **Justification**: Integrating push notifications would **alert users in real-time** about service request updates, urgent announcements, and upcoming events.
3. **Multilingual Support**
   1. **Justification**: Adding support for multiple languages would **increase accessibility**, making the app more inclusive for diverse communities. I initially wanted to implement this feature way back in task 1 but time constraints and difficulties prevented me from doing so. With more time and a better understanding of the relevant API, I would certainly implement this feature.
4. **Geolocation Services for Reports**
   1. **Justification**: Integrating **Google Maps API** or **OpenStreetMap** would allow users to **pinpoint exact locations** when reporting municipal issues, improving accuracy.
5. **Advanced Analytics Dashboard**
   1. **Justification**: A dedicated dashboard for **municipal authorities** would provide **data visualization tools** to analyze reports, track response times, and improve service efficiency.

## **Future Enhancements**

* **Improved Filtering**: Perhaps introduce more **advanced sorting and filtering** options.
* **Real-Time Notifications**: this would be beneficial as users would be notified about updates to their **service requests** and **event announcements**.

## **Conclusion**

By offering a user-friendly platform for reporting municipal issues, providing feedback, and accessing community events, the We Got You application effectively achieved its goals. An intuitive and captivating user experience was made possible by the thoughtful application of data structures, UI improvements, and intelligent suggestions. In addition to enhancing technical proficiency, this project strengthened approaches to problem-solving in recommendation systems, data structure optimization, and user interface design. The application can develop further into a useful instrument for community involvement and municipal service management in South Africa with further improvements.  
  
  
  
  
  
  
  
  
  
  
  
**Referencing list:**

* C# Corner. 2023. *C# List Tutorial - Everything You Need To Learn About List In C#*, 2 April 2023. [Online]. Available at: <https://www.c-sharpcorner.com/article/c-sharp-list/> [Accessed 12 January 2025.
* C# Corner. 2023. *TreeView Control in C#*, 6 September 2023. [Online]. Available at: <https://www.c-sharpcorner.com/article/treeview-control-in-C-Sharp/> [Accessed 12 January 2025]
* Geeksforgeeks. 2024. *Introduction to Max-Heap – Data Structure and Algorithm Tutorials*, 18 October 2024.[Online]. Available at: <https://www.geeksforgeeks.org/introduction-to-max-heap-data-structure/> [Accessed 20 January 2025].
* Geeksforgeeks. 2025. *Introduction to Tree data structure*, 2 January 2025. [Online]. Available at: <https://www.geeksforgeeks.org/introduction-to-tree-data-structure/> [Accessed 6 January 2025].
* Geeksforgeeks. 2024. *C# SortedDictionary Class*, 18 November 2024. [Online]. Available at: <https://www.geeksforgeeks.org/sorteddictionary-class-in-c-sharp/> [Accessed 10 January 2025].