DATABASE MANAGEMENT SYSTEM REPORT

ON

**Citizen Feedback System**

SUBMITTED TO THE SAVITRIBAI PHULE, PUNE FOR THE PARTIAL

FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING IN

**INFORMATION TECHNOLOGY**

BY

Atharva Sunilrao Wankhade 23373

Tanay Sham Raundale 23376

Under the guidance of

**Mrs. Sumitra Jakhete**

**Dr. Emmanuel** **Mark**

DEPARTMENT OF INFORMATION TECHNOLOGY

PUNE INSTITITE OF COMPUTER TECHNOLOGY, PUNE

Sr. No. 27, Trimurti Chowk, Dhankawadi, Pune – 43

**AY 2024 - 25**

Society for Computer Technology & Research’s

**PUNE INSTITUTE OF COMPUTER TECHNOLOGY**

Sr. No. 27, Trimurti Chowk, Dhankawadi, Pune-43

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the DATABASE MANAGEMENT SYSTEM report entitled

**CITIZEN FEEDBACK SYSTEM**

Submitted By

Atharva Sunilrao Wankhade 23373

Tanay Sham Raundale 23376

is a bonafide work carried out by them in the Second Year of Engineering (AY 2024-25) under the supervision of **Mrs. Sumitra Jakhete, Dr. Emmanul Mark** and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology).

Dr. E. Mark Dr. A. S. Ghotkar Dr. S. T. Gandhe

Mrs. S. A. Jakhete Head of the Department Principal

Lab Instructor and Mentor

**Date:   
Place: Pune**

**ABSTRACT**

In modern governance and public service delivery, collecting and analyzing citizen feedback is essential for enhancing service quality, transparency, and civic engagement. However, traditional methods such as manual registers, physical forms, or isolated digital platforms often lead to challenges like data inconsistency, lack of centralized access, delayed responses, and minimal accountability. These limitations reduce the effectiveness of public feedback mechanisms and hinder timely decision-making. To address these issues, we propose a Citizen Feedback System, a web-based platform designed to streamline the process of collecting, managing, and analyzing public feedback.

The system is developed using React.js for the frontend to ensure a responsive and intuitive user experience, while the backend is built using Node.js and Express.js, enabling efficient API handling and secure communication with the database. MySQL serves as the relational database for managing structured data such as user details, department responses, feedback entries, and analytics. The frontend is deployed on Vercel, while the backend is hosted on Render, ensuring scalability, performance, and high availability. Core features include role-based access for citizens, officials, and administrators; structured feedback submission with validation checks; status tracking of grievances; and visual reports for analyzing department performance.

This solution minimizes manual errors, enhances transparency, and promotes accountability in public service systems. Future enhancements include support for multi-language interfaces, integration with government portals, AI-driven sentiment analysis for feedback classification, and mobile app support for wider accessibility. The platform can be extended to serve multiple municipalities, urban and rural governance bodies, and can be monetized by offering advanced analytics and integration features to government and private institutions. By leveraging modern web technologies and database management principles, the Citizen Feedback System provides a structured, efficient, and citizen-centric approach to improving public administration.

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr. No | Title | Page No |
| 1 | Introduction | 6 |
| 2 | General Description | 9 |
| 3 | Specific Requirements | 12 |
| 4 | System Design | 15 |
| 5 | System Implementation | 19 |
| 6 | Conclusion | 21 |
| 7 | References | 22 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Sr. No | Title | Page No |
| 1 | Acronyms | 7 |
| 2 | Test Cases | 18 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Sr. No | Title | Page No |
| 1 | Schema Diagram | 15 |
| 2 | Flow Diagram | 17 |

**1. Introduction**

**1.1 Purpose**

The primary purpose of the **Citizen Feedback System** is to provide a centralized, transparent, and efficient platform for collecting and managing public feedback related to various civic departments such as water supply, electricity, sanitation, and more. Traditional methods of handling citizen complaints — including physical registers, phone calls, or fragmented digital tools — often suffer from inefficiencies, lack of accountability, and poor data organization. This system addresses these challenges by offering a web-based application that enables structured communication between citizens and the respective government departments.

The platform includes **two distinct login portals**: one for **citizens**, allowing them to submit feedback or complaints regarding public services, and another for **departmental administrators** (e.g., water, electricity), enabling them to view, track, and respond to the issues raised. Each department admin has access only to feedback related to their area, ensuring secure and organized data handling.

With built-in data validation and predefined input formats, the system ensures the integrity and completeness of all submitted information. It also features graphical dashboards and reporting tools to help administrators monitor feedback trends, prioritize actions, and make data-driven decisions. Ultimately, the Citizen Feedback System enhances civic engagement, promotes accountability in governance, and drives continuous improvement in public service delivery.

**1.2 Scope**

The **Citizen Feedback System** is designed for use by municipal corporations, local governing bodies, and public service departments to streamline the collection and resolution of citizen complaints and feedback. The system supports:

* **User login for citizens** to submit feedback related to public services.
* **Admin login for departmental officials** (e.g., Water, Electricity, Sanitation) to view, manage, and respond to feedback within their respective domains.
* **Structured feedback submission** through standardized forms with validation rules to ensure data integrity.
* **Real-time tracking** of feedback status to enhance transparency and accountability.
* **Graphical reports and dashboards** for administrators to visualize department-wise performance and identify common issues.
* **Role-based access control**, ensuring secure and compartmentalized access to data for users and admins.

**1.3 Definitions, Acronyms, and Abbreviations**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Term** | **Definition** |
| 1 | Citizen Feedback | Input, complaints, or suggestions provided by citizens regarding public services. |
| 2 | Department Admin | An official responsible for managing feedback related to a specific public service (e.g., water, electricity). |
| 3 | User | A citizen who registers and submits feedback through the system. |
| 4 | Complaint Tracking | |  | | --- | | The process of monitoring the status and resolution progress of submitted feedback. |  |  | | --- | |  | |
| 5 | |  | | --- | |  |  |  | | --- | | UI (User Interface) | | |  | | --- | | The visual elements of the application that users interact with. |  |  | | --- | |  | |
| 6 | API (Application Programming Interface) | A set of functions allowing interaction between frontend and backend systems. |
| 7 | DBMS (Database Management System) | Software for storing and managing data, in this case, MySQL. |
| 8 | React.js | A JavaScript library for building dynamic and responsive frontend user interfaces. |
| 9 | Node.js | A JavaScript runtime used to build scalable and fast backend applications. |

Table 1.1: Acronyms used in Course Attainment System

**1.4 Developers’ Responsibilities: An Overview**

The developers of the **Citizen Feedback System** were responsible for the complete software development lifecycle, ensuring the platform meets the functional and technical needs of both citizens and departmental administrators. Their responsibilities included:

* **Requirement Analysis:**
  + Understanding the needs of end-users (citizens) and departmental officials (e.g., water, electricity).
  + Identifying pain points in existing feedback handling systems, such as lack of transparency, poor tracking, and delayed responses.
* **System Design:**
  + Designing the database schema to handle user data, feedback submissions, department-wise classification, and status tracking.
  + Creating wireframes and UI flow for both citizen and admin interfaces using React.js.
* **Frontend Development:**
  + Developing a responsive and accessible UI with React.js for both user login (citizen feedback portal) and admin login (department dashboard).
* **Backend Development:**
  + Building RESTful APIs using Node.js and Express.js for handling feedback submission, authentication, and admin actions.
  + Ensuring secure user authentication, role-based access, and proper session handling.
  + Integrating **MySQL** to manage persistent storage of feedback, user data, and department logs.
* **Validation & Error Handling:**
  + Implementing real-time validation on feedback forms to ensure no required fields are left blank.
  + Handling backend errors gracefully and providing clear feedback to users.
* **Deployment:**
  + Hosting the **frontend** for fast and scalable user access.
  + Deploying the **backend** on **Render**, ensuring reliability and ease of API management.
* **Testing & Debugging:**
  + Conducting unit tests for individual components and APIs.
  + Performing integration and user acceptance testing to validate the complete system.
* **Future Planning:**
  + Proposing features such as multilingual support, mobile application integration, AI-based feedback analysis, and multi-municipality deployment for broader impact.

**2. General Description**

**2.1 Product Function Perspective**

The **Citizen Feedback System** is a web-based application designed to simplify and streamline the process of collecting, managing, and responding to citizen complaints related to public services such as water supply, electricity, sanitation, and infrastructure. Unlike traditional paper-based or fragmented systems, this platform offers a unified, digital environment that enables transparent communication between citizens and municipal departments.**Core Functionalities Include:**

**User Login Portal:**

Allows citizens to register, log in, and submit structured feedback or complaints related to specific departments.

**Department Admin Login:**

Enables department officials (e.g., Water, Electricity) to view, manage, and update the status of complaints relevant to their domain.

**Feedback Submission Forms with Validation:**

Ensures complete and valid entries using predefined formats and required field checks.

**Real-time Complaint Tracking:**

Allows users to monitor the status of their submissions (e.g., pending, in progress, resolved).

**Graphical Dashboards:**

Provides department-wise performance visualizations, including complaint volumes, resolution times, and common issues.

**Role-Based Access Control:**

Ensures users and admins have appropriate access to functionalities and data based on their roles.

**2.2 User Characteristics**

The **Citizen Feedback System** is designed to serve a diverse set of users involved in the feedback and grievance redressal process. Each user type has specific roles and access levels tailored to their responsibilities, ensuring usability, privacy, and efficiency.

**Primary Users:**

* **Citizens (General Users):**
  + Can register and log in to submit feedback or complaints.
  + Track the status of their submissions in real-time.
  + No technical knowledge is required; the user interface is intuitive and mobile-friendly.
  + Can optionally attach images or documents for better context.
* **Department Admins (e.g. Water, Electricity):**
  + Have access only to feedback relevant to their assigned department.
  + Can update complaint status (e.g., Pending, In Progress, Resolved) and add resolution comments.
  + Can view department-wise analytics to understand recurring issues and response efficiency.

**2.3 General Constraints**

**• Internet Dependency:**

As the Citizen Feedback System is a fully web-based application, all operations—including login, feedback submission, status tracking, and analytics—require a stable internet connection. Offline functionality is currently not supported.

**• Database Constraints (TiDB Cloud):**

The system uses TiDB Cloud, a distributed SQL database compatible with MySQL, for managing structured data like user accounts, complaints, department data, and feedback statuses.

**Key database constraints include:**

* Relational Integrity: Enforced using foreign key relationships between tables (e.g., user → feedback, feedback → department).
* Normalization: Applied to eliminate redundancy and maintain data consistency.
* Primary Keys: Every major table has a unique identifier to ensure precise data referencing.
* NOT NULL Constraints: Critical fields (e.g., user ID, complaint text, department ID) cannot be left empty.
* CHECK Constraints: Logical checks are applied to ensure fields like status codes or complaint categories remain valid.
* Indexes: Frequently accessed fields like user\_id, feedback\_id, and department\_id are indexed to improve query performance.
* ACID Transactions: TiDB supports distributed transactions, ensuring atomicity, consistency, isolation, and durability during complex operations.

**• Authentication Constraints (JWT):**

The system uses JSON Web Tokens (JWT) for secure, session-based authentication:

* Tokens include user role (e.g., citizen, admin) and expire after a set period.
* Token renewal and revocation mechanisms are in place to prevent unauthorized access.
* All sensitive actions require a valid token to maintain system integrity.

**2.4 Assumptions and Dependencies**

**Assumptions:**

* Users (citizens) are expected to have basic digital literacy—the ability to register, log in, and fill out simple online forms.
* Citizens will provide genuine and accurate feedback or complaints relevant to their local departments.
* Department Admins will regularly check the system to respond to and resolve submitted feedback in a timely manner.
* All departments involved (e.g., Water, Electricity) will coordinate and use the platform actively as part of a digitized public service initiative.

**Dependencies:**

* **Database:**

The application uses TiDB Cloud, a distributed SQL database. It must be properly configured and accessible with low latency to ensure seamless storage and retrieval of feedback, user data, and admin responses.

* **Authentication:**

Authentication is handled using JWT (JSON Web Tokens). Browser support for Local Storage is required for token handling and session management.

**3. Specific Requirements**

**3.1 Inputs and Outputs**

**Inputs:**

**User and Admin Login Credentials:**

Login details (Aadhar Number and password) for:

* + Citizens

Login details (Email and password) for:

* + Department Admins (e.g., Water, Electricity, etc.)

**Feedback/Complaint Form Data (from Citizens):**

* + Department selection (e.g., Water, Roads)
  + Description of issue
  + Location/area

**Admin Response Input:**

* + Complaint status updates (e.g., Pending, In Progress, Resolved)

**Search and Filter Inputs:**

* + Filters for viewing complaints by department

**Outputs:**

* **Admin Dashboards:**
  + Overview of total complaints resolved vs unresolved.
  + Department-wise Complaints Filtering.
* **Validation/Error Messages:**
  + Alerts for missing fields, invalid formats.
  + Unauthorized access

**3.2 Functional Requirements**

* FR1: The system must allow authenticated users (citizens, department admins, and super admin) to log in securely using JWT-based authentication.
* FR2: Citizens must be able to submit feedback or complaints through a form, including fields like department, description, location, and optional image uploads.
* FR3: The system must validate all input data on both client-side and server-side (e.g., required fields, file type, character limits).
* FR4: Department admins must be able to view complaints related only to their department, along with filter options for date, status, and location.
* FR5: The system must allow department admins to update the status of each complaint (e.g., Pending → In Progress → Resolved), along with remarks.
* **FR6:** The system must implement **role-based access control**, ensuring users can only access features permitted by their role.
* **FR7:** All complaints and user data must be stored persistently in the **TiDB Cloud database**, ensuring scalability and reliability.
* **FR8:** When submitting a complaint, users should be able to auto-fill their location using a “Use My Location” button, which captures latitude and longitude via the browser’s Geolocation API.
* **FR9**: The system must store the captured coordinates in the TiDB database along with the complaint record.

**3.3 Functional Interface Requirements**

* The **frontend (React.js)** must communicate with the **backend (Node.js/Express.js)** using RESTful APIs.
* The **frontend should be responsive**, fetching data using React’s use Effect and managing state with use State.
* JWT tokens must be **stored securely in local Storage** and verified on each protected route.
* The **frontend** must use the **Geolocation API** to capture user coordinates on complaint submission.

**3.4 Performance Constraints**

* **Geolocation Data Handling:**User-submitted coordinates (latitude and longitude) must be stored efficiently in the TiDB database.
* **Optimized Filtering:**  
  Each department admin (e.g., Water, Electricity, Waste Management) must **only access feedback submitted to their department**.
* **Data Integrity & Foreign Keys:**  
  While TiDB supports referential integrity, the system must be designed to **maintain logical consistency** without compromising on **insert/update speed**.
* **Fast CRUD Operations:**  
  The User and Admin can perform **CRUD operations** (Create FeedBack, Update status, Read dashboard data, Delete feedback).

**3.5 Design Constraints**

* The application must be built using **React.js** (frontend), **Node.js/Express.js** (backend), and **TiDB Cloud (MySQL-compatible)**
* Responsive UI design should be compatible with desktops and modern browsers.
* JWT should be used for authentication and role-based access must be maintained.

**3.6 Acceptance Criteria**

* **Users (citizens)** can successfully **log in or register**, submit feedback with details like Aadhar Card, description, and location (coordinates via a location button).
* **Department-specific Admins** can **log in securely**, view only the feedback relevant to their department.
* **Feedback submissions** must be validated (no missing mandatory fields) and **stored in TiDB Cloud Database** accurately.
* System accurately stores **location coordinates** when a user clicks the location button, and this data is visible on the admin dashboard.
* Admins can update complaint status (e.g., Resolved, Pending) and see real-time updates

**4. System Design**

**A diagram of a diagram

AI-generated content may be incorrect.4.1 ER-Model**

**4.2 Schema Description**

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure 4.1: Schema Diagram of Citizen Feedback System

Here's a detailed description of the tables in your Citizen Feedback System:

**User Table:**

Table users {

adhar\_no varchar(12) [pk]

name varchar(50)

email varchar(100) [unique]

password varchar(255)

address text

}

**Admin Table:**

Table admins {

admin\_id int [pk, increment]

dept\_id int

password varchar(255)

email varchar(50)

}

**Department Table:**

Table departments {

dept\_id int [pk, increment]

dept\_name varchar(100)

}

**Feedback Table:**

Table feedback {

fed\_id int [pk, increment]

status enum('Pending', 'Resolved', 'InProgress')

admin\_id int

dept\_id int

adhar\_no varchar(12)

feedback\_description text

fed\_date timestamp [default: CURRENT\_TIMESTAMP]

latitude decimal(9,6)

longitude decimal(9,6)

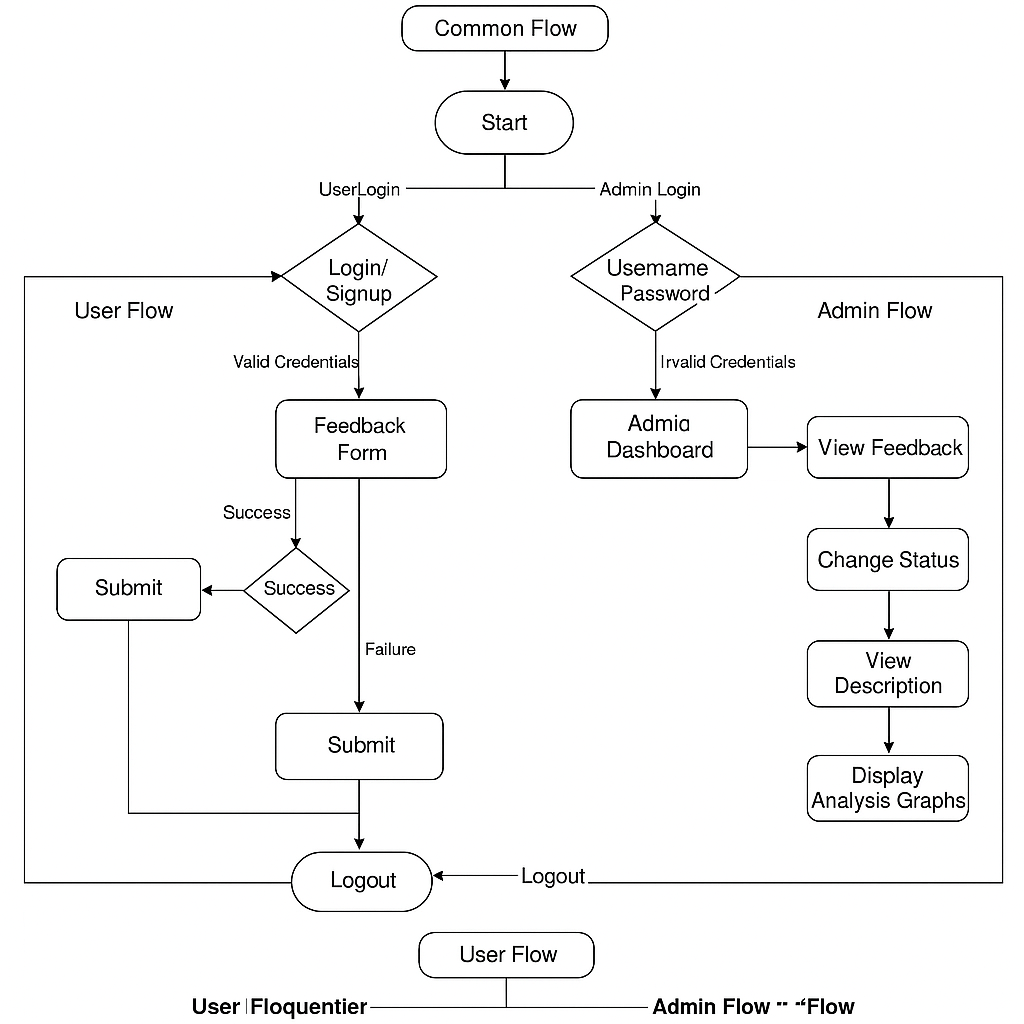
}

Ref: feedback.admin\_id > admins.admin\_id

Ref: feedback.dept\_id > departments.dept\_id

Ref: admins.dept\_id > departments.dept\_id

Ref: feedback.adhar\_no > users.adhar\_no

**4.4 System Flowchart / Activity Diagram**

View Analysis

Invalid Credentials

Resubmit



Figure 4.2: Flow Diagram

**4.5 User Interface Design**

**• Login Page:** A clean and simple UI where users and department admins can log in securely. Validation is applied for empty or incorrect inputs.

**• User Dashboard:** Users can submit feedback by selecting a department (e.g., Water, Electricity), entering issue details, and clicking a "Get Location" button to capture coordinates.

* **Admin Dashboard (Department-wise):**

Displays a **list of complaints** related only to the admin's department.

Provides **status update options** (e.g., Resolved, In Progress).

**4.7: Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC01 | Admin Verify successful login | Valid email and password | Redirect to dashboard | Pass |
| TC02 | User Verify successful login | Valid Aadhar Card Number and Password | Redirect to Feedback Form | Pass |
| TC03 | Valid Feedback form | Wrong Aadhar card number length and Miss Fields Filling | Show Invalid Message | Pass |
| TC04 | |  | | --- | |  |  |  | | --- | | Logout functionality | | Click on logout | Session terminated, redirect to login | Pass |

Table 4.1: Test Cases of Course Attainment System

**5. System Implementation**

**5.1 Hardware and Software Platform Description**

* **Hardware Requirements**:
  + Minimum 4 GB RAM, dual-core processor (for local development).
  + Stable internet connection (for web access).
* **Software Requirements**:
  + **Frontend**: React.js, HTML5, CSS3, JavaScript, Tailwind CSS.
  + **Backend**: Node.js with Express.js.
  + **Database**: TiDB Database( MySQL )
  + **Other Libraries**:
    - Axios (for API calls),
    - JWT (for authentication),
    - .env (for secure environment config),
    - React Router (for navigation),

**5.2 Tools Used**

* **Development Tools**:
  + Visual Studio Code (code editor)
  + Thunder Clients (API testing)
  + TiDB Database(MySQL)
* **Libraries/Frameworks**:
  + React.js – Frontend framework for building UI.
  + Node.js & Express.js – Backend API framework.
  + Axios – To send HTTP requests between frontend and backend.
  + JWT – For authentication and session management.

**5.3 System Verification and Testing (Test Case Execution)**

* **Testing Strategy**:
  + Unit Testing (Individual component functions)
  + Integration Testing (Frontend-backend integration)
  + Validation Testing (Data are Stort in Database are Correct)
  + Security Testing (JWT and authentication flow)
* **Test Results**:
  + Most features passed on first testing cycle.
  + Minor UI bugs resolved using developer tools.
  1. **Future Work / Extension**
* AI-Powered Feedback Categorization: Implement Natural Language Processing (NLP) to automatically categorize user complaints based on content, sentiment, and urgency.
* Public Transparency Portal: Provide a public view of resolved complaints to promote government transparency and accountability.
* Language Localization: Add support for regional languages to ensure inclusivity and better reach among local citizens.
* Mobile-friendly responsive redesign for better accessibility.

**6. Conclusion**

The Citizen Feedback System provides a structured, digital solution for efficiently managing public grievances across municipal departments. By replacing traditional, manual complaint-handling methods with an intuitive web-based platform, the system ensures transparency, traceability, and timely resolution of citizen issues. Through role-based access, real-time status tracking, and department-specific dashboards, it empowers both administrators and the public with actionable insights and responsive communication.

**7. References**

Minelli, A., & Ruffini, R. (2018). Citizen feedback as a tool for continuous improvement in local bodies. International Journal of Public Sector Management, 31(1), 46-64.

Havaeji, H., & Albadvi, A. (2019). Citizen feedback and executives: Improving government performance through citizen relationship management. International Journal of Organization Theory & Behavior, 22(3), 242-258.

Offenhuber, D. (2015). Infrastructure legibility—a comparative analysis of open311-based citizen feedback systems. Cambridge Journal of Regions, Economy and Society, 8(1), 93-112.

Lee, D., Felix, J. R. A., He, S., Offenhuber, D., & Ratti, C. (2015, July). CityEye: Real-time visual dashboard for managing urban services and citizen feedback loops. In Proceedings of the 14th International Conference on Computing in Urban Planning and Urban Management (CUPUM), Cambridge, MA, USA (pp. 7-10).