Customer Design Map for City Tours

1. Introduction

The Customer Design Map for City Tours outlines a seamless and engaging experience for travelers, from initial entry to the conclusion of their journey. This framework ensures customer satisfaction, safety, and efficiency.

2. Steps

1. Inquiry & Booking - Customer searches for city tours and books via website, app, or travel agency.
2. Confirmation - Confirmation email/SMS is sent with itinerary and QR code/ticket.
3. Arrival - Customer arrives at designated meeting point.
4. Guided Tour Experience - City tour begins with guide briefing and navigation through the itinerary.
5. Feedback & Follow-Up - Customer provides feedback post-tour. Follow-up offers or loyalty programs are shared.

3. Process

Stage	Customer Action	Business Action	Tools Used
Pre-Tour	Research, Book Tour	Provide booking platform	Website, App
During Tour	Participate, Engage	Guide, Transport	Maps, Headsets
Post-Tour	Share Feedback	Collect data, Improve	Forms, CRM

4. Entry

Customers are required to present:

- Booking confirmation (digital or printed)
- Valid ID (passport or local ID)

- Any applicable vouchers

Entry points include:

- Designated pick-up points
- Hotel lobby (if hotel pick-up included)
- City landmarks or central transport hubs

5. Precautions

Health & Safety:

- Masks recommended in closed spaces.
- Sanitizers provided.
- Emergency contact info shared at start.

Weather Preparedness:

- Umbrellas or ponchos provided in rain.
- Hydration tips for hot climates.

Security:

- Instructions on keeping personal belongings safe.
- Clear guidelines for group movements.

6. Conclusion

City tours offer enriching experiences when customer journeys are well-designed. A structured process, clear communication, safety focus, and responsive service lead to high customer satisfaction and repeat engagement.

Data Flow Diagrams and User Stories

Date: 28 June 2025

Team ID: Ltvip2025 tmid49958

Project Name: Resolve Now - Your Platform for Online Complaints

Max Marks: 4 Marks

Data Flow Diagrams (DFD) - Examples and Description

Data Flow Diagrams (DFDs) visually represent the flow of data within a system.

They show how data enters, is processed, and exits the system.

Example:

1. External Entity: User
2. Process: Submit Complaint
3. Data Store: Complaints Database
4. Output: Complaint Confirmation

Description:

The user submits a complaint, which is processed by the system and stored in a database. The system then sends a confirmation to the user. This helps in understanding system interactions and identifying potential bottlenecks.

User Stories - Examples and Description

User stories describe a feature from the perspective of the end user.

Example:

As a user, I want to file a complaint easily, so that I can report issues quickly.

As an admin, I want to track complaint status, so that I can ensure timely resolutions.

Description:

These stories help developers understand user needs and prioritize features. They are a part of Agile development practices and facilitate collaboration between stakeholders.

Precautions

1. Ensure data accuracy in DFDs.
2. Avoid overly complex diagrams; keep them understandable.
3. Maintain confidentiality of user data in design.
4. Validate user stories with actual user feedback.

Conclusion

Data Flow Diagrams and User Stories are crucial tools in system design and development. They enhance clarity, support planning, and ensure that user needs are central to the project. Using these tools effectively leads to more user-friendly and robust systems.

Project Design Phase II: Solution Requirements

Date: 28 June 2025

Team ID: Ltvip2025tmid49956

Project Name: Resolve Now - Your Platform for Online Complaints

Max Marks: 4M

1. Functional Requirements:

- User Registration and Login:

Users should be able to register, log in securely, and manage their accounts.

- Complaint Submission:

Users can submit complaints with details including type, location, and description.

- Complaint Tracking:

Users can track the status of their submitted complaints in real-time.

- Admin Dashboard:

Admins can manage complaints, assign tasks, and update statuses.

- Notifications:

Automatic updates via email/SMS when complaint status changes.

2. Non-Functional Requirements:

- Performance:

The platform should handle multiple user requests simultaneously without delay.

- Scalability:

Should support future expansion in terms of user base and data.

- Security:

All data must be encrypted, with secure login mechanisms (e.g., 2FA).

- Usability:

The interface should be intuitive and user-friendly for all age groups.

- Availability:

Platform should be operational 24/7 with minimal downtime.

3. Description:

"Resolve Now" is an online complaint management platform designed to simplify the process of reporting and resolving public grievances. The platform ensures transparency, accoun timely responses through real-time tracking and effective communication between administrators.

2. Non-Functional Requirements:

- Performance:

The platform should handle multiple user requests simultaneously without delay.

- Scalability:

Should support future expansion in terms of user base and data.

- Security:

All data must be encrypted, with secure login mechanisms (e.g., 2FA).

- Usability:

The interface should be intuitive and user-friendly for all age groups.

- Availability:

Platform should be operational 24/7 with minimal downtime.

3. Description:

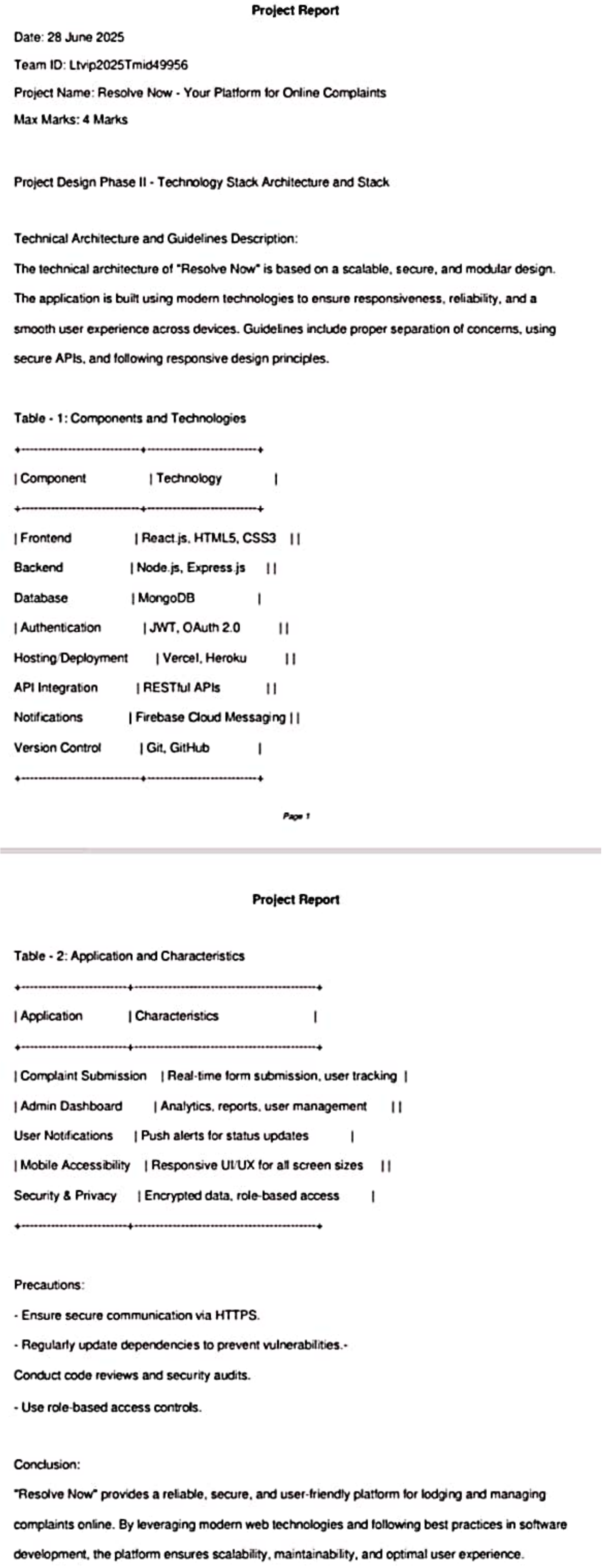
"Resolve Now" is an online complaint management platform designed to simplify the process of reporting and resolving public grievances. The platform ensures transparency, accountability, and timely responses through real-time tracking and effective communication between users and administrators.

4. Precautions:

- Ensure proper validation of user inputs to prevent security vulnerabilities.
- Backup user data regularly to avoid data loss.
- Implement proper access control to safeguard sensitive information.
- Monitor system performance and uptime to maintain service reliability.

5. Conclusion:

The project aims to create a reliable, secure, and user-centric platform for handling complaints efficiently. By addressing both functional and non-functional requirements, "Resolve Now" strives to provide an impactful digital solution for grievance redressal.



Project Report

Date: 28 June 2025

Team ID: Ltvip2025Tmid49956

Project Name: Resolve Now - Your Platform for Online Complaints

Max Marks: 4 Marks

Project Design Phase II - Technology Stack Architecture and Stack

Technical Architecture and Guidelines Description:

The technical architecture of "Resolve Now" is based on a scalable, secure, and modular design. The application is built using modern technologies to ensure responsiveness, reliability, and a smooth user experience across devices. Guidelines include proper separation of concerns, using secure APIs, and following responsive design principles.

Table - 1: Components and Technologies

Component	Technology
Frontend	React.js, HTML5, CSS3
Backend	Node.js, Express.js
Database	MongoDB
Authentication	JWT, OAuth 2.0
Hosting/Deployment	Vercel, Heroku
API Integration	RESTful APIs
Notifications	Firebase Cloud Messaging
Version Control	Git, GitHub

Table - 2: Application and Characteristics

Application	Characteristics
Complaint Submission	Real-time form submission, user tracking
Admin Dashboard	Analytics, reports, user management
User Notifications	Push alerts for status updates
Mobile Accessibility	Responsive UI/UX for all screen sizes
Security & Privacy	Encrypted data, role-based access

Precautions:

- Ensure secure communication via HTTPS.

- Regularly update dependencies to prevent vulnerabilities.-

Conduct code reviews and security audits.

- Use role-based access controls.

Conclusion:

"Resolve Now" provides a reliable, secure, and user-friendly platform for lodging and managing complaints online. By leveraging modern web technologies and following best practices in software development, the platform ensures scalability, maintainability, and optimal user experience.