

## Sprint 2 Planning Document

**Project:** Community Service App for the Government of Ontario

**Team:** PRJ666 – Team ReportEase

**Sprint Duration:** Weeks 4–6 (3 Weeks)

---

### Sprint Goal

Sprint 2 focuses on enabling residents to report public issues and empowering clerks to manage these complaints. It aims to build and test the full complaint lifecycle, including issue submission with images and GPS data, viewing reports, role-specific dashboards, and ticket escalation mechanisms.

---

### Sprint Team Members & Assigned Tasks

| Name                       | Assigned Tasks  |
|----------------------------|---|
| Vrundaben Vijaykumar Patel | CS-007: Issue Reporting Form, CS-012: View Reported Issues                        |
| Sanskar Parakhlal Pardesi  | CS-009: GPS Location Integration, CS-026: UI Customization Based on Role          |
| Nadi Aung Lin              | CS-008: Image Upload and Validation, Frontend Integration across CS-007 to CS-014 |
| Abhi Mansukhbhai Chakrani  | CS-010: Store Complaint in MongoDB, CS-013: Clerk Dashboard: Acknowledge Tickets  |
| Yash Patel                 | CS-014: High Priority Escalation & UI Testing across CS-007 to CS-014             |

---

### Sprint Scope

This sprint delivers issue reporting functionality for residents and complaint handling for clerks:

- Create and validate the complaint form with image and location
  - Enable residents to track submitted issues
  - Allow clerks to acknowledge and escalate reports
  - Store all data in MongoDB with accurate schema
  - Implement UI role-based customization
-

## Sprint Backlog Items

| ID     | Title                                | Description  | Assignee        | Effort (SP) |
|--------|--------------------------------------|--|-----------------|-------------|
| CS-007 | Issue Reporting Form                 | Submit issue with description, category, and location    | Vrunda Patel    | 5           |
| CS-008 | Image Upload and Validation          | Attach images to complaints with size/format validation  | Nadi Lin        | 3           |
| CS-009 | GPS Location Integration             | Auto/manual location capture and Google Maps integration | Sanskar Pardesi | 5           |
| CS-010 | Store Complaint in MongoDB           | Store complaints with metadata in database               | Abhi Chakrani   | 3           |
| CS-012 | View Reported Issues (Resident)      | Residents view and track reports with statuses           | Vrunda Patel    | 3           |
| CS-013 | Clerk Dashboard: Acknowledge Tickets | Clerks can acknowledge and change ticket status          | Abhi Chakrani   | 5           |
| CS-014 | High Priority Escalation             | Clerks can escalate critical complaints to admin         | Yash Patel      | 5           |
| CS-026 | UI Customization Based on Role       | Display dashboard options based on logged-in role        | Sanskar Pardesi | 3           |
|        | UI Testing                           | Ensure responsive and working UI across components       | Yash Patel      | 4           |
|        | Frontend Integration                 | Ensure responsive integration across components          | Nadi Lin        | 4           |

---

## Sprint Task Breakdown

- **CS-007:** Build a form for category, description, and optional manual location. Integrate it into the dashboard.
- **CS-008:** Allow image attachments, validate format and size before submission. Link image to complaint in DB.
- **CS-009:** Use GPS API to auto-fetch coordinates or allow manual input. Display on Google Maps preview.
- **CS-010:** Save submitted complaint with resident ID, timestamp, and media path to MongoDB.

- **CS-012:** Provide a dashboard view for residents with their submitted issues and statuses.
  - **CS-013:** Provide clerks with a view to acknowledge issues. Update issue status and trigger notification.
  - **CS-014:** Add "Escalate" option for urgent issues. Capture justification and notify admin.
  - **CS-026:** Dynamically show UI sections (buttons, pages) based on role. Hide irrelevant features.
  - **UI Testing (Yash):** Collaborate with all members to test responsiveness, layout alignment, and integration.
- 

### **Sprint Acceptance Criteria**

- Residents can submit and view issues with location and image
  - Clerks can acknowledge and escalate issues
  - MongoDB contains all data with valid schema
  - UI adapts based on logged-in role
  - Images are uploaded securely and validated
  - All pages pass responsive design testing
- 

### **Sprint Deliverables**

1. Issue Reporting Form with working image & location integration
  2. Fully functioning Clerk dashboard for issue handling
  3. Resident dashboard with personal complaint history
  4. MongoDB complaints collection with linked images and metadata
  5. Escalation system for urgent issues
  6. Role-based dynamic UI navigation & dashboard views
  7. Updated project demo deployed to Vercel
- 

### **Definition of Done (DoD)**

- Tasks committed and merged with main/dev branches
- MongoDB schema updated with proper validation
- UI tested on mobile/tablet/desktop views

- Images validated and uploaded correctly
  - Role-based routes protected and working
  - App deployed and accessible with working issue features
- 

### Sprint Risks & Mitigation

| Risk                             | Mitigation Strategy  |
|----------------------------------|--|
| GPS or Maps integration may fail | Use fallback manual input with validation                    |
| Image upload may exceed limits   | Resize client-side or restrict size via validation           |
| DB schema inconsistency          | Use centralized schema definitions and validation middleware |
| UI bugs across devices           | Assign dedicated QA role (Yash) to test multiple layouts     |

---

### Sprint Timeline

#### Week Key Activities

- 4 Start on CS-007 to CS-010; build forms, integrate MongoDB, add image validation
  - 5 Implement role-specific dashboards and UI, work on Clerk actions and escalation
  - 6 Finalize escalation, complete integration testing and deploy to Vercel
- 

### Sprint Review & Demo Goals

- Submit and view complaint as resident
- View and acknowledge issue as clerk
- Show GPS and image data being captured and displayed
- Escalate complaint and trigger admin notification
- Show MongoDB entries and working dynamic UI across roles
- Walk through commits and branches in GitHub