|  |  |  |
| --- | --- | --- |
| **CG/CSG QA report - V2** | Test start date - Dec 23, 2024  Test end date - Dec 30, 2024 | Report prepared  Dec 31, 2024 |

# **Test Scope**

The test scope covered the evaluation of the CG/CSG application based on the following core functionalities as per the Software Requirements Specification (SRS):

1. **Member Management**: Adding, viewing, and updating group member details.
2. **Work plan Management**: Creating and updating annual work plans.
3. **Social Map**: Uploading and updating demographic data.
4. **Meeting Scheduling and Conducting**: Scheduling and tracking group meetings.
5. **Disaster Management**: Assigning disaster preparedness plans.
6. **Reporting**: Viewing and analyzing CSG reports.
7. **Beneficiary Management**: Registering and tracking beneficiaries.
8. **Courtyard Sessions**: Scheduling and conducting sessions for beneficiaries.

The testing assessed the functional aspects, including form validations, data flow, mockups, and integration points.

# **Observations**

The testing revealed significant issues with the application’s functionality and its alignment with the SRS. Many of the core features outlined in the SRS are either incomplete or non-functional. This includes critical modules like member management, workplan tracking, social map updates, and meeting management. The lack of functional features greatly impacts the application’s utility and readiness for deployment.

Additionally, there are dependencies between the application’s implementation and the SRS that are problematic because the SRS itself is deficiently documented. Key workflows, validation criteria, and integration points are either undefined or inadequately described. This absence of clarity in the SRS directly affects the development process, leading to incomplete or inconsistent implementations.

Some examples of these dependencies include:

* Undefined workflows for recurring updates, such as work plans and disaster data.
* Missing guidelines for generating unique IDs for members and their backend dependencies.
* Ambiguities in the representational states of features like courtyard sessions and social maps.
* Lack of progress tracking metrics and alignment with reporting requirements.

These gaps highlight the need for a more robust and comprehensive approach to documenting and aligning the SRS with development requirements. Without this alignment, the application risks further delays and misaligned functionality.

# **QA Recommendation**

1. **Enhance Documentation**:
   * Clearly define project scope, target areas, and initial/final user base.
   * Provide detailed technical specifications, including offline/online features.
2. **Improve UI/UX Design**:
   * Finalize and validate all mockups.
   * Resolve overlapping elements and inconsistencies in the UI.
3. **Define Validation Rules**:
   * Create a comprehensive list of input requirements and validation criteria for all forms.
   * Establish unique ID generation mechanisms with backend dependencies documented.
4. **Expand Functional Coverage**:
   * Specify workflows for recurring updates, such as workplans and disaster data.
   * Clarify meeting and courtyard session handling, including notifications and dependencies.
5. **Incorporate Multilingual Support**:
   * Add English-to-Bangla transliterations for key indicators.
6. **Mockups and Reporting**:
   * Complete mockups for all modules, including reports and disaster management.
   * Define progress tracking metrics and ensure alignment with user requirements.
7. **Deployment Planning**:
   * Outline deployment and hosting strategies, ensuring alignment with client’s criteria.

# **Bug Reports**

Identified bugs can be found in this link -

### 

### **Status of the reported bugs**

* 10 issues reopened from the V1 (Smoke testing of the app).
* 36 new issues have been posted.
* Bug severity among the issues are as follows: 10 Critical, 24 High, 11 Medium and 5 Low.