

CS-3009 – SOFTWARE ENGINEERING Project Document (Iteration-0)

Team Members:

1. Muneeb-ul-Islam – i22-1289
 2. Haziq Naeem – i22-1214
 3. Ahmed Mustafa- i22-2301
-

1. Project Title

Community-Based Management System

2. Problem Statement

In many residential communities, managing daily operations, including complaint resolution, visitor tracking, event organization, announcements, and payment processing, is often inefficient and fragmented. Residents struggle to report and track complaints, security officers face challenges in managing visitor logs, and event organizers lack streamlined RSVP and notification mechanisms. Additionally, admins need better tools for automating announcements and managing maintenance dues.

The Community-Based Management System (CBMS) aims to address these issues by providing a unified platform where residents, admins, security officers, maintenance staff, and event organizers can interact seamlessly. The system will enable residents to submit complaints, register visitors, RSVP for events, receive notifications, and manage payments conveniently. Admins will have tools to track engagement, assign maintenance tasks, automate announcements, and ensure smooth operations within the community. By digitizing these essential functions, CBMS will enhance communication, improve response times, and foster a more organized and efficient community environment.

3. Envisioned Features

1. Complaint submission system
2. Complaint tracking dashboard
3. Complaint categorization by urgency
4. Complaint assignment module
5. Complaint status update functionality
6. Visitor pre-registration feature
7. Visitor check-in and check-out logging
8. Visitor arrival notifications
9. Visitor access restriction for unregistered visitors
10. Visitor log reporting system

11. Event creation module
12. Event browsing feature
13. Event RSVP system
14. Attendee limit enforcement
15. Event reminder notifications
16. Community announcements posting system
17. Push notifications for urgent notices
18. Announcement scheduling system
19. Announcement filtering by category
20. Announcement read tracking system
21. Monthly maintenance bill generation
22. Online dues viewing module
23. Online payment processing system
24. Automated payment reminders
25. Payment receipt generation

4. User Stories

User Story 1:

As a resident, I want to submit a complaint about a community issue so that it can be resolved promptly.

Acceptance Criteria:

- And I know I am done when:
 - I can fill out and submit a complaint form.
 - I receive a confirmation message after submitting the complaint.
 - My complaint appears in the complaint tracking system.

User Story 2:

As a resident, I want to track my complaint status so that I know when it will be resolved.

Acceptance Criteria:

- And I know I am done when:
 - I can see the current status of my complaint.
 - I receive notifications on status updates.

User Story 3:

As an admin, I want to categorize complaints based on urgency so that critical issues get prioritized.

Acceptance Criteria:

- And I know I am done when:
 - I can select a priority level when reviewing a complaint.
 - Complaints are sorted by urgency level in the dashboard.

User Story 4:

As an admin, I want to assign complaints to maintenance staff so that they can be addressed efficiently.

Acceptance Criteria:

- And I know I am done when:
 - I can select a maintenance staff member and assign a complaint to them.
 - The assigned staff receives a notification about the complaint.
 - The complaint status updates to "Assigned" in the tracking system.

User Story 5:

As a maintenance staff member, I want to update the status of assigned complaints so that residents stay informed about progress.

Acceptance Criteria:

And I know I am done when:

- I can mark a complaint as "In Progress," "Resolved," or "Needs Further Action."
- The complaint status updates in the tracking system.
- The resident receives a notification about the status change.

User Story 6:

As a resident, I want to register my visitor in advance so that security can allow them entry without delay.

Acceptance Criteria:

And I know I am done when:

- I can enter my visitor's details and submit a registration request.
- I receive a confirmation that my visitor has been registered.
- The security team can view the registered visitor in the system.

User Story 7:

As a security officer, I want to verify and log visitor check-ins and check-outs so that entry records are maintained.

Acceptance Criteria:

And I know I am done when:

- I can search and verify a visitor's registration upon arrival.
- I can log the visitor's check-in and check-out time.
- The system updates the visitor's status in the entry records.

User Story 8:

As a resident, I want to receive a notification when my visitor arrives so that I can approve their entry.

Acceptance Criteria:

And I know I am done when:

- I receive a real-time notification when my visitor arrives.
- I can approve or reject my visitor's entry from the app/system.
- Security receives my decision and allows or denies entry accordingly.

User Story 9:

As a security officer, I want to deny visitor entry if they are not pre-registered so that community security is maintained.

Acceptance Criteria:

And I know I am done when:

- I can check whether a visitor is pre-registered.
- If the visitor is not pre-registered, I can deny entry and log the attempt.
- The system notifies the resident that an unregistered visitor has arrived.

User Story 10:

As an admin, I want to generate a visitor log report so that security audits can be conducted efficiently.

Acceptance Criteria:

And I know I am done when:

- I can generate a visitor log report for a specific date range.
- The report includes visitor names, check-in/check-out times, and approval status.
- The report can be exported or printed for audit purposes.

User Story 11:

As an event organizer, I want to create an event with a title, date, and venue so that residents can RSVP and attend.

Acceptance Criteria:

And I know I am done when:

- I can enter the event title, date, time, and venue in the system.
- The event appears in the event listing for residents.
- Residents can RSVP for the event.

User Story 12:

As a resident, I want to browse upcoming events so that I can decide which ones to attend.

Acceptance Criteria:

And I know I am done when:

- I can view a list of upcoming events with details like date, time, and venue.
- I can filter or search for specific events.
- I can view additional event details when clicking on an event.

User Story 13:

As a resident, I want to RSVP for an event so that I receive updates and reminders.

Acceptance Criteria:

And I know I am done when:

- I can select an event and submit my RSVP.
- I receive a confirmation of my RSVP.
- I receive notifications and reminders about the event.

User Story 14:

As an admin, I want to limit the number of attendees for an event so that overcrowding is prevented.

Acceptance Criteria:

And I know I am done when:

- I can set a maximum number of attendees for an event.
- The system prevents additional RSVPs once the limit is reached.
- Residents receive a notification if an event is fully booked.

User Story 15:

As an event organizer, I want to send event reminders to attendees so that they don't forget to attend.

Acceptance Criteria:

And I know I am done when:

- I can schedule automatic event reminders before the event starts.
- Attendees receive reminders via email, SMS, or app notifications.
- The reminders include event details such as time, venue, and any important instructions.

User Story 16:

As an admin, I want to send important community announcements so that all residents stay informed.

Acceptance Criteria:

And I know I am done when:

- I can compose and send announcements to all residents.
- Residents receive the announcement via email, app, or SMS.
- The announcement appears in the community news section.

User Story 17:

As a resident, I want to receive push notifications for urgent notices so that I don't miss any updates.

Acceptance Criteria:

And I know I am done when:

- I receive a push notification when an urgent announcement is posted.
- The notification includes a link to view the full announcement.
- I can adjust notification settings based on urgency level.

User Story 18:

As an admin, I want to schedule announcements for automatic posting so that I don't have to do it manually every time.

Acceptance Criteria:

And I know I am done when:

- I can set a date and time for an announcement to be posted automatically.
- The system posts the announcement at the scheduled time.
- I can edit or cancel scheduled announcements before they go live.

User Story 19:

As a resident, I want to filter announcements based on categories (e.g., security, maintenance, events) so that I can focus on relevant updates.

Acceptance Criteria:

And I know I am done when:

- I can view announcements categorized by topics like security, maintenance, and events.
- I can apply filters to only see announcements relevant to me.
- The system remembers my selected filters for future visits.

User Story 20:

As an admin, I want to track how many residents have read an announcement so that I can measure engagement.

Acceptance Criteria:

And I know I am done when:

- I can see a count of how many residents have viewed an announcement.
- I can access engagement reports showing read statistics over time.
- I receive insights on which types of announcements get the most engagement.

User Story 21:

As an admin, I want to generate monthly maintenance bills for each resident so that they can pay their dues on time.

Acceptance Criteria:

And I know I am done when:

- I can generate maintenance bills for all residents automatically each month.
- Each resident receives their bill with details like amount due and due date.
- The system updates the billing records for tracking payments.

User Story 22:

As a resident, I want to view my pending dues online so that I can stay updated on my payments.

Acceptance Criteria:

And I know I am done when:

- I can log in and view my outstanding maintenance dues.
- I can see details such as the due amount, due date, and past payments.
- The system updates my balance in real time after a payment is made.

User Story 23:

As a resident, I want to pay my maintenance dues online so that I don't have to visit the office physically.

Acceptance Criteria:

And I know I am done when:

- I can choose an online payment method (credit/debit card, bank transfer, etc.).
- My payment is processed securely and confirmed instantly.
- My payment status updates in the system, and the due amount is adjusted.

User Story 24:

As an admin, I want to send automated payment reminders so that residents don't miss deadlines.

Acceptance Criteria:

And I know I am done when:

- The system sends reminders before the payment due date.
- Residents receive notifications via email, SMS, or app alerts.
- The reminders stop once the payment is completed.

User Story 25:

As a resident, I want to download a receipt after making a payment so that I have proof of payment.

Acceptance Criteria:

And I know I am done when:

- I can download a payment receipt immediately after making a payment.
- The receipt includes details like payment date, amount, and transaction ID.
- I can access and re-download past receipts from my payment history.

Here's a complete version of your report section with all the necessary details filled in:

5. Team Roles

Product Owner: Muneeb-ul-Islam – i22-1289

- Defines the vision and scope of the project.
- Ensures that the product meets stakeholder and user needs.
- Prioritizes backlog items and ensures smooth development flow.
- Acts as the bridge between stakeholders and the development team.

Scrum Master: Ahmed Mustafa – i22-2301

- Facilitates Agile practices and ensures Scrum framework adherence.
- Removes obstacles that hinder the team's progress.
- Organizes and moderates Scrum ceremonies (Daily Stand-ups, Sprint Planning, Reviews, and Retrospectives).
- Supports team members in resolving conflicts and improving productivity.

Scrum Team: Haziq Naeem – i22-1214

- Implements features based on the Product Owner's priorities.
 - Participates in Sprint Planning, Reviews, and Retrospectives.
 - Ensures code quality, performs testing, and commits work regularly.
 - Collaborates with the team to resolve technical challenges.
-

6. Team Agreement

Communication

- **Methods of communication:** Phone, WhatsApp, Text, Trello.
- **Response times:** Expected response time for messages is within **2 hours** during working hours (8 :30 AM - 10 PM). Urgent matters should be responded to ASAP.

Meetings

- **Meeting attendance:**
 - Weekly sprint meetings every **Sunday at 6 PM to 8 PM (Online via Google Meet)**.
 - Daily Stand-up Meetings (Short updates via WhatsApp/Trello).
 - Additional meetings if necessary.
- **Running meetings:**
 - **Structure:** Starts with a progress update, discussion of blockers, and planning for the next steps.
 - **Location:** Online (Google Meet) or in-person if needed.
 - **Note-taking:** Scrum Master records meeting minutes and updates the Trello board.
- **Meeting preparation:**
 - Team members must review assigned tasks before the meeting.
 - Any blockers or issues should be noted for discussion.

Version Control & Work Division

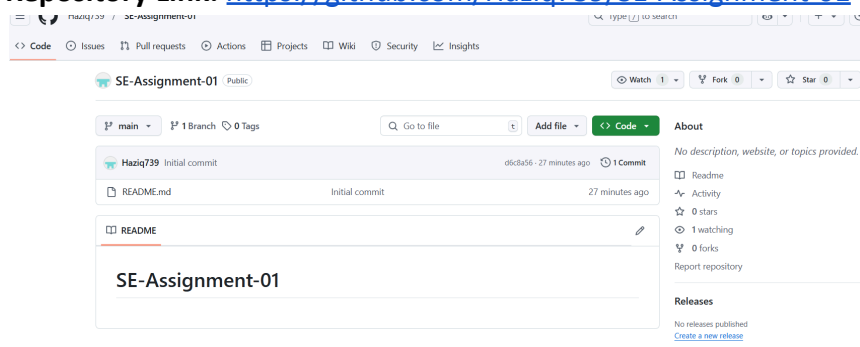
- **Version control:**
 - GitHub repository used for all commits.
- **Division of work:**
 - Tasks are assigned based on team expertise and workload.
 - Work is tracked using **Trello** (Board with columns: Product Backlog, SPrint Backlog, Done).
 - Team members update progress regularly.
- **Stakeholders:**
 - University faculty/supervisor.
 - Community residents (end-users once the Web system is done).
 - Developers & testers(Teacher Assistant).

Submission & Contingency Planning

- **Assignment submission:**
 - Final submission is due **our assigned date by teacher**.
 - Code is reviewed by the Scrum Master and Product Owner before submission.
 - Submission includes the **GitHub repo link, documentation, Trello SnapShots**.
 - **Contingency planning:**
 - If a team member is unable to contribute due to unforeseen reasons, their work will be reallocated.
 - Missed meetings must be caught up via meeting minutes and Trello updates.
 - Academic dishonesty is strictly prohibited—each team member is responsible for their assigned work.
-

7. GitHub Repository

- **Repository Link:** <https://github.com/Haziq739/SE-Assignment-01>



TRELLO SNAPSHOTS:

