

Using Crowd-Sourcing to Identify and Solve Community Problems

–User Scenarios–

Craciun Catalin

Cozma Rares

Eminovici Andrei

Fechete David

Group: 30443

Contents

1	User Scenarios	2
1.1	Scenario for Browsing Activities	2
1.2	Scenario for Reporting an Issue	2
1.3	Scenario for Selecting a Voting Option	3
1.4	Scenario for Enabling Alerts	4
2	Alternative Scenarios	5
2.1	Alternative Scenario 1 for Applying to an Activity	5
2.2	Alternative Scenario 2 for Enabling Alerts	6
3	Design Motivation	7

1 User Scenarios

1.1 Scenario for Browsing Activities

Starting Point in the UI:

- The user lands on the homepage of the platform, where the "Browse activities" button is prominently displayed.

User Actions:

- The user clicks on the "Browse activities" button.
- The interface redirects to the Activities page, displaying a list of available activities in card format.
- The user scrolls through the activity cards to find one of interest.

System Reactions:

- Upon clicking "Browse activities," the system navigates to the Activities page.
- Activity cards are displayed in a grid layout, each showing the activity's name, date, and a brief description.

Success/Error Cases:

- **Success:** The Activities page loads successfully, and activity cards are visible.
- **Error:** If the Activities page fails to load, an error message is displayed (e.g., "Unable to load activities. Please try again later.").

Design Motivation:

- The prominent placement and clear labeling of the "Browse activities" button follow the principle of visibility, ensuring users can easily locate key actions. The use of activity cards provides a clean and organized way to present information, aligning with best practices for content display.

1.2 Scenario for Reporting an Issue

Starting Point in the UI:

- The user navigates to the "Report Issue" page via the navigation menu.

User Actions:

- The user fills in the "First Name," "Last Name," and "Email" fields.
- The user describes the issue in the "Issue" text area.
- The user clicks the "Submit" button to report the issue.

System Reactions:

- As the user fills out each field, real-time validation highlights any missing or incorrectly formatted inputs.
- Upon clicking "Submit," the system validates the form and, if successful, displays a confirmation message (e.g., "Your issue has been reported successfully.").

Success/Error Cases:

- **Success:** The form submits successfully, and a confirmation message is shown.
- **Error:** If required fields are missing or improperly formatted, error messages prompt the user to correct them.

Design Motivation:

- Real-time validation and clear feedback upon submission adhere to usability principles by reducing user errors and providing immediate guidance. The straightforward form layout ensures users can complete tasks efficiently without unnecessary complexity.

1.3 Scenario for Selecting a Voting Option

Starting Point in the UI:

- The user is on the "Vote" page, where voting options are displayed as distinct buttons.

User Actions:

- The user reads the purpose of the vote in the descriptive section.

- The user selects an option (e.g., "More funds to the metro") by clicking the corresponding button.
- The user clicks the "Submit Vote" button to finalize their choice.

System Reactions:

- Upon selecting an option, the button changes color to indicate selection.
- After clicking "Submit Vote," a confirmation message is displayed (e.g., "Your vote has been recorded. Thank you!").

Success/Error Cases:

- **Success:** The vote is recorded, and the user receives confirmation.
- **Error:** If there's an issue recording the vote, an error message informs the user (e.g., "Unable to record your vote. Please try again later.").

Design Motivation:

- Clear labeling and visual feedback for selected options enhance user confidence in their actions. The confirmation message follows best practices by assuring users that their input has been successfully processed.

1.4 Scenario for Enabling Alerts

Starting Point in the UI:

- The user navigates to the "Alerts Settings" page via the settings menu.

User Actions:

- The user locates the toggle switch labeled "Automatic alerts for your issues."
- The user clicks the toggle to enable alerts.
- The user adds a specific area (e.g., "Marasti") by clicking the "Add area" button and entering the area name.

System Reactions:

- Toggling the switch changes its state (e.g., from off to on) and displays a confirmation message (e.g., "Alerts for your issues have been enabled.").

- Adding an area triggers a modal where the user enters the area name, and upon submission, the new area appears in the list with a success message.

Success/Error Cases:

- **Success:** Alerts are enabled, and the new area is added successfully.
- **Error:** If adding an area fails, an error message is shown (e.g., "Unable to add the area. Please try again.").

Design Motivation:

- The use of toggle switches provides a familiar and accessible way for users to manage settings. Confirmation messages reinforce user actions, ensuring clarity and reducing uncertainty.

2 Alternative Scenarios

2.1 Alternative Scenario 1 for Applying to an Activity

Starting Point in the UI:

- The user is on the Activities page, viewing a specific activity card.

User Actions:

- The user hovers over the activity card to reveal additional options.
- The user clicks the "Apply Now" button instead of "Apply."
- A confirmation modal appears asking for final confirmation.
- The user confirms the application, and the activity status updates to "Applied."

System Reactions:

- Hovering reveals the "Apply Now" button.
- Clicking opens a confirmation modal.
- Upon confirmation, the system updates the application status and notifies the user.

Success/Error Cases:

- **Success:** The application is submitted successfully, and the status updates.
- **Error:** If the application fails, an error message is displayed.

Design Motivation:

- Providing a confirmation step reduces accidental applications, enhancing user control and satisfaction.

2.2 Alternative Scenario 2 for Enabling Alerts

Starting Point in the UI:

- The user is on the "Alerts Settings" page.

User Actions:

- The user decides to disable alerts for survey results.
- The user clicks the toggle switch for "Automatic alerts for survey results," turning it off.
- The system immediately reflects the change with a message: "Survey alerts have been disabled."

System Reactions:

- The toggle switch changes state, and the confirmation message is displayed.

Success/Error Cases:

- **Success:** Alerts are successfully disabled, and the user is notified.
- **Error:** If disabling alerts fails, an error message is shown.

Design Motivation:

- Immediate visual and textual feedback ensures users are aware of the changes they have made, enhancing transparency and trust.

3 Design Motivation

The design of the user interface prototype is grounded in established usability principles and best practices to ensure an intuitive and efficient user experience. Key motivations include:

- **Visibility of System Status:**
Clear feedback mechanisms (e.g., confirmation messages, visual indicators) inform users about the results of their actions, aligning with Nielsen's usability heuristics.
- **Match Between System and the Real World:**
The use of familiar controls like buttons, toggle switches, and progress bars corresponds to users' real-world experiences, reducing the learning curve.
- **User Control and Freedom:**
Features such as undo options, confirmation modals, and clear navigation paths empower users to manage their interactions confidently.
- **Consistency and Standards:**
Consistent labeling and control placement across different pages enhance predictability, making the interface easier to navigate.
- **Error Prevention and Recovery:**
Real-time validation and clear error messages help prevent user errors and provide guidance for recovery, ensuring a smooth user journey.
- **Aesthetic and Minimalist Design:**
A clean and organized layout avoids clutter, allowing users to focus on essential tasks without distractions.
- **Accessibility:**
Incorporating tooltips, hover descriptions, and clear labels ensures the interface is accessible to users with varying levels of expertise and from different backgrounds.
- **Flexibility and Efficiency of Use:**
Providing multiple pathways to achieve the same goal (e.g., different user scenarios) caters to diverse user preferences and enhances overall efficiency.

By adhering to these principles, the interface design facilitates a user-friendly environment where users can effortlessly accomplish their tasks, leading to higher satisfaction and engagement.