Pentominoes

HeartSteal
Use-Case Specification: Manage Notifications

Version 1.0

Revision History

Date	Version	Description	Author
18/07/2025	1.0	Provide use case specification for Manage Notification	Hoàng Minh Giang

Table of Contents

Use-Case Specification: Manage Notifications

1.   Use-Case Introduction

1.1   Brief Description

This use case allows a user to configure their notification preferences within the app. Users can enable or disable different types of notifications, such as new photo alerts, friend requests, or activity updates. The goal is to provide control over when and how users receive updates from the system.

1.2   Primary Actor

The User

2.   Flow of Events

2.1   Basic Flow

1.   The user navigates to the Settings screen from the main screen.
2.   The user selects the "Notifications" option.
3.   The system displays the current notification preferences.
4.   User adjusts the settings, such as:
    o   Turning notifications on/off globally.
    o   Enabling/disabling specific types (e.g., "New Post", "Friend request", "New Message").
5.   User taps "Save"
6.   The system validates and updates the stored preferences.
7.   A confirmation is shown to the user (e.g., "Your changes have been saved").

2.2   Alternative Flows

2.2.1   *Failure to Retrieve Preferences (Step 3)*

1.   The system fails to fetch current settings due to a server timeout or offline status.
2.   The system displays an error: "Cannot load settings. Try again later."
3.   The user may retry or exit. If retry succeeds, return to Step 3.

2.2.2   *Save Operation Fails (Step 5)*

1.   The system encounters a backend error during the save operation.
2.   The user sees: "Changes could not be saved. Retry?"
3.   The user may retry Step 5, or exit and retry later.

2.2.3   *User Cancels Without Saving (Step 5)*

1.   The user makes changes but leaves the screen without saving
2.   The system shows a prompt: "Discard unsaved changes?
3.   The user confirms or returns to editing.

2.2.4   *User Close Application (Step 5)*

1.   The user close the application without saving
2.   The system discard all changes

3.   Special Requirements

●   All changes must sync with the backend and persist across sessions and devices.
●   Users must have the ability to review their current settings anytime.
●   The system must comply with local laws (e.g., GDPR for notification opt-outs).
●   Changes should reflect within 5 seconds across all devices.

4.   Preconditions

●   The user must be logged in.
●   Network connectivity must be available to fetch or save preferences.

5. Postconditions

- Preferences are stored and active.
- Future notifications adhere to updated settings.
- The user is shown confirmation or informed of any failure.

6. Prototype