|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [notification\_manager.py]** | | |
| **Summary***:*  The notification\_manager.py file is responsible for handling various types of notifications within Razda Market, including email, SMS, and in-app notifications. It integrates with email and SMS APIs (using SMTP and Twilio) and a MySQL database for in-app notifications. Additionally, it includes functionality to retrieve, view, and mark notifications as read, offering a centralized notification management system to improve user engagement and communication. | | |
| ***Processes*** | | |
| * **Email Notifications** | **Sends email notifications to users through SMTP, including subject and message content.** | |
| * **SMS Notifications** | **Utilizes Twilio API to send SMS notifications to users, leveraging Twilio's authentication and sending services.** | |
| * **In-App Notifications** | **Sends in-app notifications to users by storing notifications in a MySQL database. These notifications can be viewed in the user interface.** | |
| * **Retrieve Notifications** | **Retrieves notifications for a specified user, with the option to filter for unread messages only.** | |
| * **Mark Notification as Read** | **Allows marking specific notifications as read, updating their status in the database to improve notification tracking** | |
| * **Main CLI Interface** | **Provides a command-line interface for sending, retrieving, and marking notifications as read, streamlining management tasks.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Environment Variables** (dotenv): Loads configuration for SMTP, Twilio, and database credentials. | * L**ogging File** (notification\_manager.log): Logs all notification activities, errors, and critical events. | |
| * **Twilio and SMTP Servers**: For SMS and email notifications. | * **SMTP Server**: Sends email notifications through configured email service. | |
| * **Database** (razda\_market\_db): Retrieves user data for in-app notifications. | * **Twilio API**: Sends SMS notifications using Twilio API and authentication. | |
| **Expected input into file:** | | **Expected output from file:** |
| **Notification Parameters**: Contact information, user ID, severity level, message content, etc.  ‘’’python  to\_email, subject, message, user\_id, channel, severity  ‘’’  **Database Credentials**: For accessing in-app notification records in razda\_market\_db.  **User Notification Requests**: Command-line input specifying the notification type and recipient. | | **Notifications**: Sends notifications (email, SMS, in-app) based on the specified channels.  ‘’’python  self.send\_email(to\_email, subject, message)  ‘’’  **Logs**: Logs notification send status and any errors encountered.  **Database Updates**: Marks notifications as read or stores new notifications in in\_app\_notifications table. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Notification Delivery: Ensures successful delivery of notifications via the selected channel (email, SMS, or in-app), with logs for successful and failed attempts.** |  |  | | --- | | **● Database Connection Management: Maintains a stable connection with the MySQL database for in-app notifications, with error handling for connection issues.** |  |  | | --- | | **● Twilio and SMTP API Configuration: Loads and verifies Twilio and SMTP credentials for SMS and email functionality, with fallback for unconfigured services.** |  |  | | --- | | **● User Interaction with CLI: Accepts command-line inputs for managing notifications, including commands to send, retrieve, and mark notifications as read.** |  |  | | --- | | **● Notification Status Management: Allows marking notifications as read in the database, ensuring users can track their read/unread notifications.** |  |  | | --- | | **● Role-Based Message Logging: Logs different notification levels and includes a summary of any critical issues or failed deliveries for administrative review.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| Contribution to the entire Project:  The **notification\_manager.py** file plays a crucial role in Razda Market's communication strategy by centralizing notification management. It improves user engagement by ensuring that notifications are promptly sent through various channels. This file's robust configuration for email, SMS, and in-app notifications enhances both user experience and administrative oversight, enabling Razda Market to maintain effective communication across its platform. | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: Initial setup for email, SMS, and in-app notification functionalities; integrated with SMTP and Twilio APIs; connected to MySQL for in-app notifications. | | |