**Notification Bot BRD (Business Requirements Document)**

**Project Overview**

**Bot Name:** (To be decided)

**Objective:** To create a notification bot that allows admins to create events and send notifications to subscribed users. The bot will also allow users to register for events and receive reminders before the event.

**Functional Requirements**

**Admin Features**

1. **Event Creation**: Admin can create new events by providing details like:
   * Event title
   * Event description
   * Date and time
   * Location
2. **Broadcast Notifications**: When a new event is created, the bot sends a notification to all subscribed users.

**User Features**

1. **Event Notifications**: Users receive notifications for:
   * New events.
   * Reminders (1 day, 3 hours, and 1 hour before the event).
2. **Event Registration**: Users can register for events by clicking a button or sending a command.
3. **Event Details**: Users can view details of upcoming events.

**Reminder System**

* Notifications sent at:
  + 1 day before the event.
  + 3 hours before the event.
  + 1 hour before the event.

**Subscription Management**

* Users can subscribe or unsubscribe from receiving event notifications.

**Non-Functional Requirements**

1. **Platform Support**: The bot will work on Telegram.
2. **Scalability**: The bot should handle up to 10,000 users without performance issues.
3. **Reliability**: Ensure notifications are delivered promptly and accurately.

**Technical Requirements**

1. **Programming Language**: Python.
2. **Frameworks**:
   * Telegram Bot API.
   * Scheduler library (e.g., APScheduler).
3. **Database**:
   * Use PostgreSQL to store user subscriptions, event details, and registration information.
4. **Hosting**:
   * Use cloud services like AWS, Heroku, or Google Cloud for deployment.

**User Interface**

1. **Admin Panel**: Accessible via bot commands (e.g., /create\_event, /list\_events).
2. **User Interaction**:
   * Start command (/start) for new users.
   * Commands like /register, /unsubscribe for user actions.

**Timeline**

1. **Week 1**: Define bot structure and create a basic framework.
2. **Week 2**: Implement event creation and notification system.
3. **Week 3**: Add user registration and reminder features.
4. **Week 4**: Testing and deployment.

**Risks and Mitigation**

1. **Risk**: Notification delays.
   * **Mitigation**: Use robust scheduling libraries and reliable hosting services.
2. **Risk**: High user load.
   * **Mitigation**: Optimize database queries and scale hosting resources.

**Success Metrics**

1. 90% of notifications are delivered on time.
2. 80% of users actively engage with event notifications.
3. Admins find the event creation process intuitive and efficient.

**Next Steps**

1. Confirm bot name and branding.
2. Finalize hosting platform.
3. Begin development as per the timeline.

  