**Header and Personalized Greeting Integration - WireFrame**

**WireFrame Design -**

**1. Header and Personalized Greeting Integration**

**Header Component:**

* **Position:** Fixed at the top of the screen, spanning the full width.
* **Background Color:** Dark gray to black, consistent with the monochrome color scheme.
* **Elements:**
  + **Logo:** Positioned on the left side, using a minimalist design, preferably a simple icon or the app's initials.
  + **Greeting Message:** Centered horizontally, dynamically updating based on the time of day. For example, "Good Morning, [User's Name]".
  + **Profile Icon:** Positioned on the right side, a circular avatar image representing the user.
  + **Notification Bell:** Next to the profile icon, with a badge indicator showing the number of unread notifications.
* **Font:** Sans-serif, consistent with the app’s typography.
* **Responsive Design:** Adjusts layout for different screen sizes. For smaller screens, elements are appropriately resized and spaced to maintain clarity.

**Detailed Wireframe Design: Header and Personalized Greeting Integration**

**Header Component**

**Overall Structure:**

* **Position:** Fixed at the top of the screen, spanning the full width.
* **Background Color:** Dark gray to black, maintaining the monochrome color scheme.
* **Height:** Approximately 60-80px, depending on screen size for a balanced appearance.

**Elements:**

1. **Logo:**
   * **Position:** Left side of the header, aligned vertically centered.
   * **Design:** Minimalist design, such as a simple icon or the app's initials (e.g., "PN" for Personalized Nutrition).
   * **Size:** Around 40-50px in height to maintain prominence but not overwhelm other elements.
   * **Padding:** Approximately 10-20px from the left edge to provide some breathing room.
   * **Interaction:** Clicking the logo redirects the user to the home screen.
2. **Greeting Message:**
   * **Position:** Centered horizontally in the header, aligned vertically centered.
   * **Content:** Dynamically updates based on the time of day. For example, "Good Morning, [User's Name]".
   * **Font Size:** Approximately 18-24px, large enough to be readable but not dominating.
   * **Font Weight:** Medium to bold to make the text stand out against the dark background.
   * **Dynamic Logic:**
     + Use JavaScript Date object to get the current hour.
     + Update greeting based on the hour:
       - Morning (5 AM - 11 AM): "Good Morning, [User's Name]"
       - Afternoon (12 PM - 5 PM): "Good Afternoon, [User's Name]"
       - Evening (6 PM - 9 PM): "Good Evening, [User's Name]"
       - Night (10 PM - 4 AM): "Good Night, [User's Name]"
   * **Accessibility:** Ensure the greeting message is accessible with proper ARIA labels.
3. **Profile Icon:**
   * **Position:** Right side of the header, aligned vertically centered.
   * **Design:** Circular avatar image representing the user.
   * **Size:** Approximately 40-50px in diameter, consistent with the logo's height.
   * **Padding:** Approximately 10-20px from the right edge and 10px spacing from the notification bell.
   * **Interaction:** Clicking the profile icon opens the user's profile or settings page.
   * **Accessibility:** Add ARIA labels for screen readers.
4. **Notification Bell:**
   * **Position:** Next to the profile icon, aligned vertically centered.
   * **Design:** Simple bell icon with a badge indicator showing the number of unread notifications.
   * **Size:** Approximately 30-40px in height.
   * **Badge Indicator:**
     + **Position:** Top-right corner of the bell icon.
     + **Size:** Small red circle with a number inside, indicating unread notifications.
     + **Font Size:** Small, around 12px, to fit within the badge.
     + **Color:** Red background with white text for the badge.
   * **Interaction:** Clicking the bell icon opens the notifications dropdown or panel.
   * **Accessibility:** Ensure the notification bell and badge indicator are accessible with ARIA labels.

**Font:**

* **Font Family:** Sans-serif, consistent with the app’s typography.
* **Font Size:**
  + Greeting Message: 18-24px
  + Badge Indicator: 12px
* **Font Weight:** Medium to bold for the greeting message to ensure readability against the dark background.

**Responsive Design:**

* **Adjustment for Smaller Screens:**
  + **Logo:** May resize to around 30-40px in height.
  + **Greeting Message:** Font size may reduce to 16-20px, and the message may truncate if necessary (e.g., "Good Morning").
  + **Profile Icon:** May resize to 30-40px in diameter.
  + **Notification Bell:** May resize to 25-30px in height, with the badge indicator adjusting accordingly.
* **Layout Adjustments:**
  + **Stacking Elements:** On very small screens, consider stacking the profile icon and notification bell below the greeting message or utilizing a collapsible menu.
  + **Spacing:** Adjust padding and margins to maintain a clean and uncluttered layout.
  + **Hide Less Important Elements:** If screen real estate is extremely limited, consider hiding the greeting message or notification bell and providing access via a dropdown or slide-out menu.

**Visual Summary**

* **Header (Dark gray to black background, 60-80px height)**
  + **Left:** Logo (40-50px height, 10-20px padding)
  + **Center:** Greeting Message ("Good Morning, [User's Name]", 18-24px font size, medium-bold weight)
  + **Right:**
    - **Notification Bell:** (30-40px height, badge indicator with red background and white text)
    - **Profile Icon:** (Circular, 40-50px diameter, 10-20px padding)
* **Responsive Adjustments:**
  + Resize elements on smaller screens
  + Potentially stack or hide elements to maintain clarity and usability

**Backend Components for Header and Personalized Greeting Integration**

To implement the header and personalized greeting integration, several backend components are needed to ensure smooth operation, secure data handling, and dynamic content delivery. Here's a detailed breakdown of the necessary backend components:

**1. User Information Storage**

**Database Schema:**

* **Users Collection/Table:**
  + userId: Unique identifier for the user (e.g., UUID)
  + firstName: User's first name (string)
  + lastName: User's last name (string)
  + email: User's email address (string, unique)
  + profileImage: URL of the user's profile image (string)
  + timeZone: User's time zone (string, optional)
  + notifications: Array of notification objects, each containing:
    - notificationId: Unique identifier for the notification
    - message: Notification message
    - isRead: Boolean indicating if the notification has been read
    - timestamp: Timestamp of when the notification was created

**2. API Endpoints**

**User Profile Endpoint:**

* **Purpose:** Fetch user profile information including the user's name and profile image.
* **Endpoint:** GET /api/user/profile
* **Response:**

json

Copy code

{

"userId": "uuid",

"firstName": "John",

"lastName": "Doe",

"profileImage": "https://example.com/profile.jpg",

"timeZone": "America/New\_York"

}

**Notifications Endpoint:**

* **Purpose:** Fetch unread notifications for the user.
* **Endpoint:** GET /api/user/notifications
* **Response:**

json

Copy code

{

"notifications": [

{

"notificationId": "uuid",

"message": "You have a new message",

"isRead": false,

"timestamp": "2024-06-27T12:34:56Z"

},

{

"notificationId": "uuid",

"message": "Your weekly report is ready",

"isRead": false,

"timestamp": "2024-06-26T08:21:33Z"

}

]

}

**Mark Notification as Read Endpoint:**

* **Purpose:** Mark a notification as read.
* **Endpoint:** POST /api/user/notifications/mark-read
* **Request Body:**

json

Copy code

{

"notificationId": "uuid"

}

**3. User Authentication**

**JWT Authentication:**

* **Purpose:** Securely authenticate users and authorize access to their profile and notifications.
* **Components:**
  + **JWT Tokens:** Issue JWT tokens upon successful login, storing them in HTTP-only cookies or local storage.
  + **Middleware:** Verify the JWT token for protected routes, ensuring the user is authenticated.

**4. Time Zone Handling**

**User Time Zone:**

* **Purpose:** Determine the appropriate greeting message based on the user's local time.
* **Implementation:**
  + **Frontend Logic:** Use the browser's time zone if the user's time zone is not stored in the backend.
  + **Backend Storage:** Optionally store the user's time zone in the database and provide it in the user profile response.

**5. Data Security and Privacy**

**Secure Communication:**

* **HTTPS:** Ensure all API endpoints are accessible over HTTPS to protect data in transit.
* **Encryption:** Encrypt sensitive data stored in the database, such as user profile information.

**Data Privacy Compliance:**

* **Compliance:** Adhere to data privacy regulations such as GDPR or CCPA, ensuring user data is handled responsibly.

**6. Performance Optimization**

**Caching:**

* **Purpose:** Reduce load times by caching user profile information and notifications.
* **Implementation:** Use server-side caching mechanisms like Redis to store frequently accessed data temporarily.

**Efficient Queries:**

* **Purpose:** Optimize database queries to minimize response times.
* **Implementation:** Index relevant fields such as userId and isRead to speed up lookups.

**7. Error Handling**

**Robust Error Handling:**

* **Purpose:** Gracefully handle errors and provide meaningful feedback to the frontend.
* **Implementation:**
  + **API Responses:** Standardize error responses with appropriate HTTP status codes and messages.
  + **Logging:** Log errors for monitoring and debugging purposes.

**Summary**

The backend components for the header and personalized greeting integration include user information storage, API endpoints for fetching user profile and notifications, user authentication using JWT, time zone handling, data security and privacy measures, performance optimization techniques, and robust error handling. These components work together to provide a seamless, personalized experience for users, ensuring their data is secure and interactions are smooth.