Feature name : Notification System
Developer name : Kevin Dinh
Date developer submitted : 03/07/2023
Reviewer name : Garrett Tsumaki
Date review completed : 03/07/2023

Summary:

Overall this design seems like it will run with adjustments and mending of typos, although it lacks some major requirements and does not take many cases into account. Criteria that would likely take place in the front end code have been excluded, as they are primarily dependent on implementation of this document, but display of visual features should be present in the wireframes. Functionally, there seems to be a surplus of database calls that should be attended to, although it appears to be in a near-ready state, if given a couple days more time of work.

Unmet Requirements:

- **BRD Reference-Business Rules:**
    - Rules not accounted for:
        - “Notification on the website will be displayed via a popup banner and will fade away over a short period of 8 seconds.”
        - “Group Notifications can be expanded on in List Menu to view each individual notification packaged in.”
- **BRD Reference-Display On-Site Notification:**
    - Success Criteria not accounted for in wireframe:
        - “Notification Banner pops up and slowly fades away after a duration.
- **BRD Reference-Select Notification Content Type**:
    - Failure Cases not accounted for:
        - “Notification based on type chosen do not appear”
- **BRD Reference-Display Notification Banner:**
    - Success Criteria not accounted for in wireframe:
        - “Notification Banner pops up and disappears after a set duration.”
- **BRD Reference-Expand Notification Menu:**
    - Success Criteria not accounted for in wireframe:
        - “Can scroll through list”
            - Unclear from wireframe. Recommend adding a scroll bar somewhere.
- **BRD Reference-Display Group Notifications**:
  The queue does not start on arrival of the first notification as mentioned in BRD (Display Group Notifications precondition), but rather is constant, and occurs every 1 minute, no matter if a notification has arrived or not.
    - Success Criteria not accounted for:
        - “Group Notification successfully displays how many notifications are contained”
    - Failure Criteria not accounted for:
        - “Group Notification pings but still displays each individual one”

- **BRD Reference-Expand Group Notifications**:
  Completely unmentioned. Context changed in the design doc relative to how the BRD describes it. *Note: Could be accounted for in front-end, but there is no explanation how the group notifications will be differentiated in the back end
    - Business Rules not accounted for:
        - “Group Notifications can be expanded on in List Menu to view each individual notification packaged in.”
    - Success Criteria not accounted for:
        - “Notification clearly identifies how many notifications are contained within”
        - “Clicking grouped notification expands into a list of each individual notification”
    - Failure Criteria not accounted for:
        - “Group Notification does not show up on Notification Menu”
        - “Notifications are not grouped as they should be”
- **BRD Reference-Auto-Deletion at Capacity**:
  Notifications are hidden and not deleted. Could be considered the same effect, but should be discussed.
    - Questionable Business Rules:
        - “A List Menu for notifications will hold 50 notifications before deleting the oldest ones.”
- **BRD Reference-Clear All Notifications:**
    - Success Criteria not accounted for in wireframe:
        - “Popup occurs to confirm clearing all notifications“
            - Not shown in wireframe.
- **BRD Reference-Dismissal of Individual Notification:**
    - Success Criteria not accounted for in wireframe:
        - “Option to clear appears after hovering notification”
        - “Clicking “clear” dismisses individual notification”

Document Clarity recommendations:

- **Title page**:
    - Date is wrong.
- **Revision History**:
    - looks like this was copied from a previous document? Please update to be relevant to the current document.
- **Requirements To Satisfy**:
    - The specificity of requirements from the BRD should be maintained, and can just be made more succinct when moved here in order to not need repeated checks of the BRD by the reader.
- **HLD-Create notification**:
    - Why would deletion create a notification? Deletion deletes their account along with all of their account settings.
    - What is “Verify Request”? Do you mean Authorize? Do you mean sanitize? What are you verifying?
    - Should possibly be separated into two different functionalities, as “Create Notification” and “Send notification” as this does not just create a notification, but also sends it to the user. (see later for renaming instead)
- **HLD-Alter Notification Settings**:
    - Split into two diagrams - “View Notification Settings” and “Alter Notification Settings”. This would add more clarity, as these are two different use cases.
- **HLD-Notification Menu**:
    - This must be split up. The arrows for filtering and updating the view are nearly unreadable. This can be split up into “Show notification menu”, “Filter Notification View”, and “Reset Notification View”. If not that, there needs to be conditions on the arrows to understand how things are meant to connect. The current diagram makes it seem as if it is required that the user filter the notifications, and beyond that is generally hard to read.
- **All HLD**:
    - Why are they numbered if the numbers do not associate with BRD requirements or anything else? Please remove them.
- **General**:
    - Consistent capitalization and naming absent. Please fix.
    - Spellcheck diagrams please.
- **LLD-Notification is Created, but no email…**:
    - Should also include the text/sms instance, or could be altered to have both in one diagram. Otherwise we are not testing for it.

Design Recommendations:

- **HLD-Create notification**:
    - If every time a notification is created, the text/email is sent, this could perhaps just be one DB call. Downside being the addition of extra DAL code with singular functionality.
- **HLD-On-Site Notifications**:
    - Query only new Notifications, since you have stored the “previous iteration” according to the view diagram.
    - If a query is coming from the user view, it needs to be authorized through the authorization service. Possible personal info leak could happen in the exposing of another user’s notifications.
- **HLD-Account Change Notification**:
    - Do not go to the DB. This information should be stored in the encoded JWT, which can be decoded by our backend to hand the email back to the email service without going to the DB. This would further reduce DB calls without much more work. Possible the JWT could be tampered with, but is very unlikely, as they would need our key.
- **HLD-Notification Menu**:
    - Every time the user loads notifications it queries the data store? This may be too easily spammable, and may need safeguards. Could cache this information, as mentioned later to see reduced DB load.
- **HLD-Dismiss Notification**:
    - No arrow between “Update view” and “User dismisses Notification”. It is generally implied that a user can do any action more than once, as long as they satisfy the given preconditions.
- **General**:
    - Caching is likely ideal here for performance. On login, A user’s preferences, and current notifications can be cached in the Notification service, only requiring writes to the DB instead of reads every minute. With this, this would also no longer require a DB call for preferences on every notification added, as this information is less likely to change, and can be stored in the cache as well. When any notifications are added or settings changed, they can be symmetrically changed in the DB and in cache, so that on restart the information is safe, but then also the response time is much faster. If that is too much, add a flag in the for a user if they have received a new notification and only then grab from DB then rather than on every refresh of notification list. Of course, this would require much more work, but performance may be significantly better.
- **Notification Relational Tables**:
    - Move CellProvider and CellPhoneNumber to its own table. Best practice to not touch the main tables when possible. If this is only used for notification settings, perhaps just store them in notification settings.
    - There cannot be more than one Primary key, there must only be one. If this is meant to imply that DateCreated and UserId are both part of the PK (Primary Key), then it should not be the case, as it could be possible for a single user to

get two notifications at the exact same time. Instead use NotificationId as the Primary key, otherwise there is no reason for the column.
    - UserNotifications should have a foreign key on UserId, as we wouldn't want it to be possible to have remnants of user data left in our system after account deletion.
- **LLD-Another Service…**:
    - "CreateNotification" is misleading, and I would go looking for another function that also sends notification if I were writing code that needed to call the service. Perhaps "SendNotification"? Unsure of exact naming, but Create is too specific while inaccurate to what it is doing.
    - If "CreateNotification" is always also getting the delivery method, both can be retrieved from the DB at the same time to reduce DB calls. Naming would be hard, but it should improve performance. If not that, then the caching I mentioned before would reduce DB calls without this.
- **LLD-Notification Banner:**
    - Comparing the payload on the user end means sending 50 string notifications up to the view level to compare what they have there. This is more network overhead than necessary if the initial request just queried for notifications after the last id in the previous iteration instead.
    - This should be going to the Authorization Service as well. Need to authorize access for grabbing notifications as this may contain personal information on a user.
- **LLD-User alters Notification Settings…**:
    - This should be going to the Authorization Service as well. Need to authorize access for grabbing notifications as this may contain personal information on a user.
- **User is able to navigate to the Notification Menu View:**
    - This should be going to the Authorization Service as well. Need to authorize access for grabbing notifications as this may contain personal information on a user.
- **User is able to navigate to Notification Menu View and its… failure cases**:
    - DB call seems to be incomplete. Also it queries for "viewable" which is not a valid column. Also this would grab from every user, not just the user it should be querying for.
- **LLD-User dismisses notification on Notification Menu**:
    - Each notification has its own id in the database, so it should delete based on that for more accuracy, as there could be an instance where there are more than one notifications created with the same timestamp stored in the database.
- **LLD-User clears… and User dismisses…**:
    - These are the same, but "Read" is not a column in the DB, and therefore this should be removed. Or, add Read to the DB, and implement the required changed to this doc.

Test Recommendations:

- If both “CreateNotification” DB calls were to be compressed, then there would have to be tests for the checking of either of them failing resulting in an overall failure, or specific outcomes thought out to the combination.
- Prototype the caching method and see if there is a significant enough improvement in response times. Beyond this, if implemented, symmetry would have to be verified between the database and the cached notifications + cached notification settings.
- If a “Read” functionality were to be added to the Notification system, it would require further success and failure cases, where failure cases would likely be if a read object stays read, or that by default a notification is set to not read.