News Feed and Notifications with Feedback for Universities

Team 4

Phase 2: Project Proposal Presentation

Project Problem Domain

The goal of this project is to develop a front-end interface for a university news feed and notification system that allows users to receive news updates, announcements, and notifications from different university departments and provide feedback. The interface will be designed to support students, faculty, and administrative staff, each of whom will have different usage patterns and requirements. Mock data will be used to simulate the backend, but the primary focus is on the usability and interface design.

Types of Users

The types of users who will interact with this application are as follows:

1. **Students:** Undergraduate and graduate students will use the application to stay informed about university announcements, course updates, events, and deadlines.
2. **Faculty:** Professors and academic staff will use the system to communicate with students and stay informed about department and university-wide news.
3. **Administrative staff:** University administrators will use the interface to post announcements and monitor feedback from users.

Knowledge Level Required:

- **Application domain:** Basic knowledge of university systems, courses, events, and notifications. The application is tailored for users who are familiar with the university's structure and internal communication methods.
- **Device usage:** Users are expected to have moderate proficiency in using mobile phones, tablets, and desktop computers. The interface should be intuitive and easy to navigate for users with varying technical expertise.

Special User Requirements:

- The interface will include accessibility features such as larger text, high-contrast themes, and screen reader compatibility for visually impaired users.

Challenges and Motivation

The primary challenge faced by users in the university domain is information overload. With multiple communication channels (emails, bulletin boards, social media, etc.), it is difficult to keep track of all the updates. Additionally, students and staff might miss important announcements or deadlines due to inconsistent or delayed notifications.

Motivation: This project aims to centralize all university-related updates into a single, easy-to-navigate interface, ensuring that students, faculty, and staff receive timely and relevant information. It will also allow users to provide feedback on the information they receive, enabling departments to gauge the effectiveness of their communications.

Context of Use

- The application will be primarily used in quiet or moderately noisy environments (classrooms, libraries, dorm rooms). However, it should be usable in busier environments like cafeterias or outdoor areas.
- Users might be interrupted frequently by their other tasks (classwork, meetings, etc.), so the interface should allow for quick access to important information.
- Time-sensitive elements are critical—students need to receive notifications promptly for class cancellations or deadline extensions.

Existing Applications

Currently, users rely on a combination of email systems, messaging apps (such as WhatsApp), and the university's website or Learning Management System (LMS) for updates. However, these platforms are not centralized, leading to scattered information and missed notifications.

Cooperation with Other Tools

Yes, users will likely use this interface alongside other tools such as email clients or calendar apps. Therefore, the design will include options to integrate with external calendars and set reminders directly from the notification feed. Notifications will also be accessible through push notifications on mobile devices to ensure users do not miss important updates.

Phase 2a: Task Analysis Examples

Task 1: Receiving Course-Related Notifications

- **User:** Undergraduate student
- **Description:** A student logs into the application to receive updates on their enrolled courses. This includes lecture cancellations, new assignments, and important announcements from professors.
- **Motivation:** The student needs to stay up-to-date with course changes to avoid missing deadlines or important class sessions.
- **Context:** The task is often performed during breaks between classes or when the student receives a notification. The environment can range from quiet areas like libraries to noisy public places.

Task 2: Posting an Announcement as an Administrator

- **User:** Administrative staff
- **Description:** An administrator logs into the system to post an announcement regarding an upcoming university event. The announcement needs to reach all students and faculty.

- **Motivation:** The administrator wants to ensure maximum visibility of important information related to university-wide events.
- **Context:** The task is typically performed in an office setting, but the system should allow flexibility for mobile use if required.

Phase 2b: Management Schedule

Each team member will rotate as Team Manager for equal intervals. The rotation schedule is as follows:

- **First batch (21.10.2024 - 10.11.2024):** Cezar Naghi
- **Second batch (11.10.2024 - 1.12.2024):** Diana Vaida
- **Third batch (02.12.2024 - 22.12.2024):** Pablo Garcia
- **Third batch (23.12.2024 - 13.01.2025):** Cristian Scoropan

Phase 3: Task Analysis

This section provides detailed descriptions of 12 tasks specific to the "News Feed and Notifications with Feedback for Universities" project. Each task follows the Task-Centered User Interface Design approach and satisfies the criteria outlined for the project.

Task 1: Viewing Course Announcements

- **Starting Point:** The user (student) is logged in and sees a notification on the dashboard about a new course announcement.
- **User:** Undergraduate student
- **Steps:**
  1. The student logs into the application and views the dashboard.
  2. The student sees a notification about a new course announcement.
  3. The student clicks on the notification to open the announcement.
  4. The student reads the announcement details to understand the assignment or course update.
- **Motivation:** The student wants to stay informed about course assignments to submit them on time.
- **Context:** Typically performed in a study area (library, home) using a laptop or mobile phone.

Task 2: Providing Feedback on a Lecture Cancellation

- **Starting Point:** The user (student) receives a notification that a lecture is canceled.
- **User:** Undergraduate student
- **Steps:**
  1. The student receives a push notification about the lecture cancellation.
  2. The student taps the notification to open the details about the cancellation.
  3. The student reads the message and clicks on the "Provide Feedback" link.
  4. The student types feedback requesting study materials and submits it.
- **Motivation:** The student wants to ensure they remain on track with the course even though the lecture was canceled.
- **Context:** Performed quickly during a break or before the next lecture, often on a mobile device.

Task 3: Posting a University Event Announcement

- **Starting Point:** The user (administrator) logs into the system and navigates to the news feed section to post an announcement.
- **User:** University administrator
- **Steps:**
  1. The administrator logs into the system and navigates to the announcements section.
  2. The administrator clicks on "Create New Announcement."
  3. The administrator fills out the event details, including the date, time, location, and any necessary registration links.
  4. The administrator submits the announcement, making it visible to all students.
- **Motivation:** The administrator wants to inform students about an upcoming event and ensure high attendance.
- **Context:** Usually performed during office hours in a quiet environment using a desktop computer.

Task 4: Subscribing to Department Notifications

- **Starting Point:** The user (faculty member) logs into the system and views the available notification channels.
- **User:** Faculty member
- **Steps:**

1. The faculty member logs into the system and accesses the "Notification Settings."
2. The faculty member browses the available channels for department-specific notifications.
3. The faculty member selects the desired departments (e.g., Research Office, IT Department) and subscribes.

- **Motivation:** The faculty member wants to receive updates relevant to their work in specific departments.
- **Context:** Performed in an office on a desktop computer, typically during the workday.

Task 5: Customizing Notification Preferences

- **Starting Point:** The user (student) logs in and navigates to the settings section to adjust notification preferences.
- **User:** Graduate student
- **Steps:**

1. The student logs into the application and navigates to the "Notification Preferences" section.
2. The student selects which types of notifications they want to receive (e.g., academic, extracurricular).
3. The student saves their changes and returns to the dashboard.

- **Motivation:** The student wants to focus only on notifications that are directly relevant to their academic progress.
- **Context:** Performed at home or in a study environment using a laptop or tablet.

Task 6: Searching for Old Notifications

- **Starting Point:** The user (student) is logged in and uses the search bar to find a past notification.
- **User:** Undergraduate student
- **Steps:**

1. The student logs in and navigates to the notifications section.
2. The student clicks on the search bar and types keywords related to the notification (e.g., "assignment deadline").
3. The student reviews the search results and selects the relevant notification.

- **Motivation:** The student wants to check the details of a previously announced deadline to confirm its accuracy.
- **Context:** Typically done in a quiet study environment using a laptop.

Task 7: Reviewing Feedback from Students

- **Starting Point:** The user (professor) logs in and opens the feedback section.
- **User:** Professor
- **Steps:**
  1. The professor logs in and navigates to the feedback section.
  2. The professor reviews feedback left by students in response to recent announcements or course content.
  3. The professor replies to feedback or takes action if needed (e.g., clarifying information, updating the announcement).
- **Motivation:** The professor wants to ensure that students understand the course materials and provide any necessary clarifications.
- **Context:** Typically done during office hours on a desktop computer.

Task 8: Dismissing a Notification

- **Starting Point:** The user (student) logs in and views a list of notifications.
- **User:** Graduate student
- **Steps:**
  1. The student logs in and views the notifications list.
  2. The student identifies a notification they do not need (e.g., an event they are not attending).
  3. The student dismisses the notification to clear it from the dashboard.
- **Motivation:** The student wants to avoid clutter and focus on more relevant notifications.
- **Context:** Performed quickly on a mobile phone, often while walking between classes.

Task 9: Viewing Notifications on a Mobile Device

- **Starting Point:** The user (student) opens the app on their mobile device to view notifications.
- **User:** Undergraduate student
- **Steps:**
  1. The student opens the app on their mobile device.
  2. The student scrolls through the list of recent notifications to check for important updates.
  3. The student taps on any notification that requires further attention.

- **Motivation:** The student wants to stay informed on the go without having to log into a desktop version of the app.
- **Context:** Performed in between classes or during a commute on a mobile device.

Task 10: Sending a Notification to All Students

- **Starting Point:** The user (administrator) selects the "Send to All Students" option from the notification center.
- **User:** University administrator
- **Steps:**
  1. The administrator logs into the notification system.
  2. The administrator selects the option to send a message to all students.
  3. The administrator writes the message and selects the "Send" option.
- **Motivation:** The administrator wants to communicate an urgent university-wide update (e.g., weather alert, campus closure).
- **Context:** Typically performed during working hours on a desktop in an office.

Task 11: Opting Out of Notifications for Specific Events

- **Starting Point:** The user (student) is logged in and navigates to the event notification preferences section.
- **User:** Graduate student
- **Steps:**
  1. The student logs into the app and navigates to the notification settings.
  2. The student browses through event notifications and opts out of certain event categories (e.g., sports events).
  3. The student saves the changes and returns to the dashboard.
- **Motivation:** The student prefers not to receive event notifications they are not interested in.
- **Context:** Performed in a quiet environment using a laptop or mobile device.

Task 12: Receiving Real-Time Exam Schedule Updates

- **Starting Point:** The user (student) logs into the application and enables real-time updates for exam schedules.
- **User:** Undergraduate student
- **Steps:**
  1. The student logs into the app and accesses the exam schedule section.

2. The student toggles the setting to receive real-time updates for any changes in the exam schedule.
3. The student reviews a push notification when an exam date is modified.

- **Motivation:** The student wants to stay informed about any last-minute changes to the exam schedule.
- **Context:** Often performed in a study environment using a mobile phone or laptop.

Phase 5: Scenarios for User Tasks

Task 1: Viewing Course Announcements

Scenario A: Student Accessing New Course Announcements

- **Starting Point:** The user is logged into the application and has received a notification on the dashboard.
- **UI Elements Used:** Notification alert icon, dashboard link, course announcements section.
- **System Reaction:** Clicking the notification opens a list of course announcements, displayed as cards for easy reading.
- **Success/Error Cases:** Success is achieved when the announcement is fully readable and accessible. Error case: if the network fails, an error message is displayed with a "Retry" option.
- **Rationale:** This design aligns with the task-centered approach by simplifying access to critical updates.

Scenario B: Student Accessing Past Course Announcements

- **Starting Point:** The user navigates to the course announcements section directly from the menu.
- **UI Elements Used:** Menu icon, course announcements link, filter and search functions.
- **System Reaction:** The system loads past announcements, allowing the user to filter by date or course.
- **Success/Error Cases:** Success when the student locates the announcement; error if no matching announcements are found, prompting a message "No announcements found."
- **Rationale:** Filtering options streamline access to specific announcements, enhancing usability for time-sensitive information.

Task 2: Providing Feedback on a Lecture Cancellation

Scenario A: Student Providing Feedback on Notification

- **Starting Point:** The user taps a notification about a canceled lecture.
- **UI Elements Used:** Notification alert, feedback button, text input field.
- **System Reaction:** A feedback dialog opens with an input box and submit button.
- **Success/Error Cases:** Success occurs when the feedback is submitted; an error is shown if the field is left blank, prompting the user to enter text.
- **Rationale:** By focusing on quick feedback, the system supports timely student communication needs.

Scenario B: Student Reviewing Feedback Options

- **Starting Point:** The student views past feedback and navigates to the feedback section.
- **UI Elements Used:** Notification history, feedback option link, text input.
- **System Reaction:** The system displays recent feedback related to the course, with options for "Like" or additional feedback.
- **Success/Error Cases:** Success is a smooth input and submit process; error if connection fails, retry option appears.
- **Rationale:** The design enables convenient feedback, enhancing student engagement and response.

Task 3: Posting a University Event Announcement

Scenario A: Admin Posts Event Announcement on Desktop

- **Starting Point:** Admin navigates to the "Create New Announcement" page.
- **UI Elements Used:** Text fields for event title, description, date picker, and submit button.
- **System Reaction:** The announcement posts, visible to all users under "Events."
- **Success/Error Cases:** Success when event posts without error; if date is invalid, prompt appears.
- **Rationale:** Structured input fields guide accuracy and ensure complete information is posted.

Scenario B: Admin Posts Quick Update on Mobile

- **Starting Point:** Admin selects "Quick Post" option from the mobile dashboard.
- **UI Elements Used:** Text field, submit button.
- **System Reaction:** The announcement displays on all user dashboards.
- **Success/Error Cases:** Success when announcement posts; error if connection drops, allowing draft save.
- **Rationale:** Quick posting supports flexibility for mobile use, ideal for brief, urgent announcements.

Task 4: Customizing Notification Preferences

Scenario A: Student Adjusts Notification Settings for Courses

- **Starting Point:** The user accesses "Notification Preferences" from the settings menu.
- **UI Elements Used:** Toggle buttons, category filters.
- **System Reaction:** Notifications adjust according to preferences, showing a confirmation.
- **Success/Error Cases:** Success with feedback confirming update; error if update fails, retry appears.
- **Rationale:** Toggle settings enable intuitive control over notification types, meeting user preferences.

Scenario B: Student Enables Real-Time Updates for Exams

- **Starting Point:** User selects real-time updates option under exam notifications.
- **UI Elements Used:** Switch toggle, confirm dialog.
- **System Reaction:** A push notification preview confirms changes.
- **Success/Error Cases:** Success if user receives preview notification; error if update fails, showing prompt.
- **Rationale:** Real-time settings empower students to track crucial updates, enhancing academic performance.

Task 5: Customizing Notification Preferences

Scenario A: Student Selects Relevant Notification Types

- **Starting Point:** The student accesses the "Notification Preferences" from the settings menu.
- **UI Elements Used:** Toggle switches, category filter options.

- **System Reaction:** Notifications adjust in real-time to the student's selection, and a confirmation message appears when settings are saved.
- **Success/Error Cases:** Success when preferences are saved without issues. An error case might occur if preferences fail to save due to a connection issue, prompting a "Retry" option.
- **Rationale:** The toggle switches provide clear, quick customization options, aligning with best practices for user control and preference settings.

Scenario B: Student Adjusts Preferences for Exam Updates Only

- **Starting Point:** The student navigates to "Notification Preferences" with the intent to receive only exam-related updates.
- **UI Elements Used:** Toggle switches, filter options, save button.
- **System Reaction:** The application saves preferences, and the dashboard updates to reflect only selected notifications.
- **Success/Error Cases:** Success when preferences are applied successfully, providing a streamlined dashboard view. If an error occurs, the student receives a prompt to retry.
- **Rationale:** Allowing users to customize notifications enhances relevance and reduces unnecessary alerts, focusing only on essential updates.