From Clutter to Clarity: Redefining Notifications in Discord

Team Dragalge

Nellie K. Cordova, Roshen Jagajeevan, Luis Ochoa, & Brandon Watts
ncordova7@gatech.edu, rjegajeevan3@gatech.edu, lochoa8@gatech.edu, & bwatts38@gatech.edu

Abstract—This project aims to improve user engagement on Discord, a leading platform for community engagement and communication. Focused on refining the notification system, the project addresses challenges faced by a diverse user base, including gamers, educators, and professionals. By employing a combination of user interviews, surveys, and heuristic evaluations, we identified essential enhancements, particularly the need for an adaptive and customizable notification management system. Three prototypes—Smart Modes, Content Filters, and PriorityHub—were designed. Evaluative measures revealed PriorityHub as the optimal solution due to its effective management of notifications through user-specified priorities. The results of this project provide a systematic approach and highlight the significance of user-centric design in creating features that address varied needs, thereby making digital communication more inclusive and efficient.

1 INTRODUCTION

In today's digital landscape, where online platforms are pivotal in fostering community engagement, communication, and collaboration, the challenge of managing notifications has become increasingly significant. Discord, a widely used communication platform designed for creating and joining communities ranging from gaming groups to educational classes, is at the forefront of this challenge. This project focuses on redesigning Discord's notification system to cater to its diverse user base, including professionals, hobbyists, and social groups. Our aim is to develop a system that not only efficiently manages notifications but also tailors them to individual user needs, significantly

enhancing the overall experience on Discord. By addressing these challenges, we strive to transform Discord into an even more versatile and user-friendly platform for its millions of users worldwide.

The current one-size-fits-all approach to notifications often leads to a cluttered and overwhelming experience, with important messages getting lost amid less relevant ones. This redesign seeks to introduce a more intuitive and customizable notification system, allowing users to prioritize communications based on personal and professional needs, thereby making their Discord experience more enjoyable and productive.

The need for this redesign is underscored by common scenarios faced by Discord users, such as a student who needs to concentrate on academic alerts during study hours or a remote worker who must prioritize professional notifications during work. These examples highlight the desire for a notification system that aligns with individual preferences while enhancing the ability to stay focused and connected.

2 NEEDFINDING PLAN

The needfinding plan for this project is designed to collect detailed insights into user experiences, preferences, and challenges with the current interface. To achieve this, we will employ a three-pronged approach consisting of user interviews, surveys, and heuristic evaluations of the existing interface.

2.1 User Interviews

Our first needfinding activity involves conducting detailed interviews with Discord users. We aim to engage with approximately five participants, each session lasting around 20 minutes. Participants will be recruited from our personal networks and Discord communities, with an emphasis on ensuring a diverse range of users in terms of age, profession, and Discord usage patterns. The interviews will explore participants' current experiences with Discord notifications, challenges faced, and desired improvements. We will prepare a set of open-ended questions to encourage participants to share their thoughts and experiences freely. See *Appendix: Needfinding User Interview Questions* for specific questions.

2.1.1 *Potential Biases*

The primary bias in this activity may stem from the participants' desire to provide socially desirable responses or their relationship with the interviewer, which could influence the candor of their feedback.

2.2 Surveys

The second activity will involve distributing surveys to a broader audience to gather quantitative and qualitative data on users' experiences and preferences regarding Discord's notification system. We aim to reach approximately twenty participants, expecting each to spend around 10 minutes on the survey. The survey, to be designed on PeerSurvey, will be shared through social media and with the Georgia Tech student community. It aims to gauge how often and what kinds of notifications users get, measure their satisfaction, and identify any desired features for an improved notification system. See *Appendix: Needfinding Survey Questions* for specific questions.

2.2.1 *Potential Biases*

Survey responses may be influenced by the ambiguity of questions or participants' interpretation of them. Also, the survey might attract respondents with stronger opinions skewing the results. Additionally, there is a risk of non-response bias, where the views of those who choose not to participate could differ significantly from those who do.

2.3 Heuristic Evaluation

Our third needfinding activity involves conducting a heuristic evaluation of Discord's current notification system. We will assess the interface using predefined heuristics to pinpoint usability issues and potential improvement areas. We plan to employ the following design principles and heuristics:

- **Visibility of System Status:** This principle assesses how effectively the system communicates the status of notifications, ensuring users understand which alerts are active, muted, or customized for clear awareness of their notification settings.
- **User Control and Freedom**: This principle suggests that users should have control over the system and be able to freely navigate and interact with it. We

will assess how easily users can adjust notification preferences and navigate settings.

- **Flexibility and Efficiency of Use**: The system should accommodate both inexperienced and expert users, allowing them to tailor frequent actions to their needs. We will explore the customization options and shortcuts available for managing notifications.
- **Recognition Rather Than Recall:** Minimize the user's memory load by making objects, actions, and options visible. We will assess how the notification system presents options and information in a way that reduces the need for users to remember information from one part of the dialogue to another.

2.3.1 *Potential Biases*

The evaluators' own experiences and expectations might influence their judgment, potentially overlooking issues that real users might face or overemphasizing minor problems.

2.4 Bias Mitigation

To address the biases inherent in each approach, our needfinding plan cross-checks findings from all three methods to ensure accuracy. For example, we validate insights gained from user interviews against data from surveys and results from heuristic evaluations. This ensures a well-rounded understanding of user needs by blending the detailed qualitative insights from interviews with the wide-ranging quantitative data from surveys and the specialized expertise from heuristic evaluations. Although some biases, such as selection bias in choosing participants, might still occur, the diverse nature of our methods is designed to lay a solid foundation for effectively refining Discord's notification system.

3 NEEDFINDING RESULTS

Following the comprehensive needfinding plan, our team embarked on a series of activities aimed at gathering insights into user preferences, challenges, and behaviors. These activities included user interviews, surveys, and heuristic evaluations of the existing system. This section details the execution of these

activities and the high-level results obtained, providing a foundation for the insights that will guide the redesign process.

3.1 User Interview Findings

From the findings of five user interviews, we observed diverse experiences and preferences regarding Discord notifications. The participants were primarily in their late twenties, with two being under twenty-five. A recurring theme was the use of Discord for gaming and socializing. Users commonly reported issues with lag and the overwhelming volume of notifications in large servers, which often led to muting channels and servers. There was a notable demand for more customizable notification settings.

Regarding desired features and changes, users expressed a preference for a simple, clean, and relevant notification system akin to Facebook's. They sought global settings that could allow for prioritizing posts from specific individuals and highlighted synchronization issues across different devices. Some users desired a centralized method to quickly check important notifications without sifting through all messages in each server. The ability to mute specific tags or posts containing certain words was also favored.

Challenges included missing important notifications due to an excess of server alerts or because of muting Discord entirely in device settings. Among additional insights, there was a request for notifications customizable to various times of the day and the ability to control notification volume separately from Discord's other sound settings.

In summary, the feedback indicates a need for a more intuitive notification management tool within Discord. Users are seeking a way to stay informed without being overwhelmed by irrelevant notifications. Enhanced customization, improved syncing between devices, and smarter notification settings would significantly enhance the user experience.

For more details see Appendix: N*eedfinding User Interview Analysis and Results.*

3.2 Survey Findings

The survey collected insights from twenty-one respondents, focusing on their experiences with Discord notifications. Participants ranged widely in age, with

a majority falling within the 18-19 age bracket. The primary uses of Discord cited by respondents encompassed a blend of gaming, socializing, education, and professional work, highlighting the platform's versatility.

Participants expressed mixed satisfaction levels with the current notification system, with ratings spanning from very unsatisfied to satisfied. A notable fraction admitted to missing important messages due to the high notification volumes, signifying a potential area for improvement. The types of notifications deemed most valuable varied, with direct messages and server mentions often highlighted. Suggestions for a redesigned notification system emphasized the need for more granular control, the ability to prioritize notifications, and the integration of smart notification features tailored to user activity or time of day.

Overall, the feedback shows a great demand for enhancements in the notification system Discord. Users have a wide range of challenges, from forgetting to turn notifications on, to wanting more control over notification sounds, to the difficulty of managing alerts from multiple servers or distinguishing between diverse types of notifications. A common theme is the desire for more granular control and customizing alerts for specific posts or comments. There is also a need for better integration with device features like Apple's Focus modes, and suggestions for new functionalities like a "gaming mode" to prioritize certain notifications or a study mode to mute non-essential alerts. Among the users that offered additional comments the ideas were innovative. For instance, requesting AI-generated summaries of notifications. There were also calls for better notification distinctions, customizable sounds for diverse types of notifications, and easier access to default notification settings.

3.3 Heuristic Evaluation Findings

The heuristic evaluation of Discord's notification system, informed by user scenarios and personas, offered a detailed assessment of its usability and user experience. Our analysis identified strengths and areas for improvement in the current design, highlighting key opportunities for enhancing the system.

3.3.1 *Strengths:*

- **Visibility of System Status**: Discord effectively uses visual cues, like red dots and numbers, to indicate new notifications, helping users quickly identify unread messages.
- **User Control and Freedom**: The system allows users to mute entire servers or specific channels, offering a basic level of customization over the notifications received.
- **Flexibility and Efficiency of Use**: Customization options for individual channels enable users to prioritize certain types of notifications, enhancing the user experience for those with specific needs.

3.3.2 *Weaknesses:*

- **User Control and Freedom**: The current muting options lack granularity, limiting users' ability to fine-tune their notification experience to their preferences. There is a need for more nuanced settings that allow users to mute specific types of notifications within channels.
- **Error Prevention and Recovery**: The system could provide better feedback and guidance for users when configuring notification settings, reducing the risk of unintentional notification suppression, and improving overall usability.
- **Recognition Rather Than Recall**: The reliance on users to remember their notification settings across different servers and channels can lead to confusion. A more centralized and intuitive overview of all notification preferences could enhance usability.

3.3.3 *User scenarios and personas*

This evaluation was enriched by incorporating user scenarios and personas, including a college student struggling to keep track of essential academic notifications amid a sea of messages, and a remote worker facing challenges due to inadvertently muted channels. These personas underscored the pressing need for a notification system that is not only more intuitive and adaptable but also centered around the specific needs and preferences of its users.

3.4 Insights Gained

1. **Demand for Personalization**: There is a strong desire among users for more personalized notification settings, allowing for greater control over which notifications are received and how they are presented. Users expressed a need to filter notifications not just by type (e.g., direct messages, server mentions) but also by content relevance, time of day, and personal priority levels.
2. **Notification Overload**: The challenge of managing a high volume of notifications, sometimes running into hundreds per day, is a significant issue for users. This overload can lead to important messages being missed, indicating a need for a smarter system that can help users prioritize and manage their notifications more effectively.
3. **Integration with User Lifestyle**: Users expressed a desire for notification settings that can adapt to their current activities, such as "gaming mode" or "study mode," which would prioritize or mute notifications accordingly. This reflects a broader need for the notification system to be more context-aware and adaptable to the user's lifestyle.

3.5 Raw Results

Raw data containing notes on user interview, participants' responses to the survey, and an in-depth heuristic analysis are included in the following appendices: *Appendix: Needfinding User Interview Notes, Appendix: Needfinding Survey Responses,* and *Appendix: Needfinding Heuristic Analysis* for reference.

4 BRAINSTORMING

4.1 Brainstorming Plan

Our brainstorming plan for redesigning Discord's notification system will adopt a hybrid approach, combining individual, asynchronous idea generation with collective, synchronous discussions to accommodate our team's diverse backgrounds and geographic distribution. The chosen approach leverages the strengths of both asynchronous and synchronous brainstorming, allowing for flexible participation while fostering real-time collaboration, idea refinement,

and ensuring broad participation. All brainstorming sessions will be thoroughly documented, including the rationale behind prioritized ideas.

4.1.1 *Bias Mitigation*

To foster an inclusive and productive brainstorming environment, our plan strategically addresses common biases. By enabling team members to independently propose ideas during the asynchronous phase, we reduce **conformity bias**, ensuring that the variety and uniqueness of ideas are not overshadowed by dominant group opinions. This phase also avoids **production blocking** by allowing simultaneous idea submissions, eliminating the bottleneck of one-at-a-time contributions. Lastly, to combat **evaluation apprehension**, ground rules for synchronous sessions promote a respectful and open environment, encouraging full participation and creativity from all team members.

4.2 Brainstorming Results

Our brainstorming session yielded a range of creative ideas aimed at enhancing user experience through improved personalization and control in Discord. We will highlight the session's key insights and introduce three selected design alternatives for further development, each accompanied by a brief textual prototype.

4.2.1 *Structure*

1. **Asynchronous Ideation**: Team members independently propose ideas on a shared platform, fostering diverse input without immediate peer influence.
2. **Synchronous Sessions**: Following the asynchronous phase, the team engages in live discussions to expand on and integrate the ideas generated asynchronously, focusing on identified key areas.
3. **Idea Categorization and Prioritization**: Ideas are organized and evaluated based on relevance, impact, and feasibility to identify the most promising concepts for the redesign.

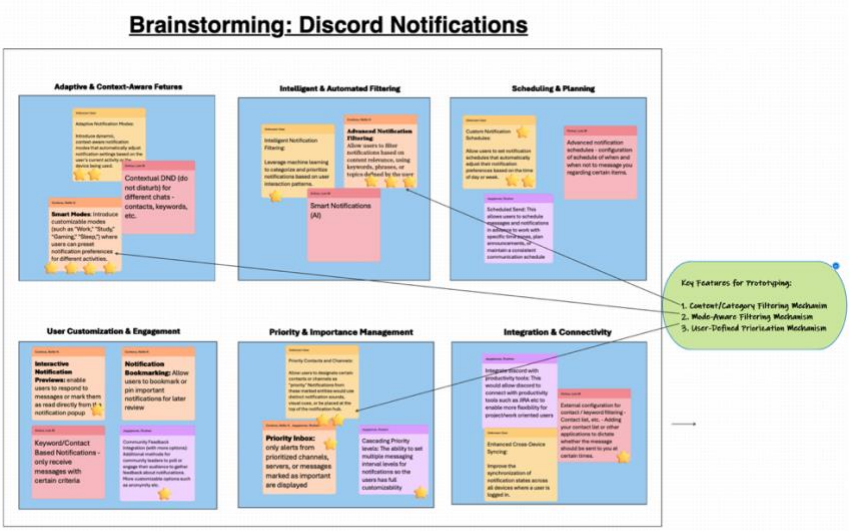


Figure 1—Group brainstorming session showcasing categorized ideas and selected key features for prototyping.

4.2.2 *Generated Ideas*

Overall, 20 ideas were generated. The ideas proposed include:

- Adaptive notification Modes
- Contextual Do not Disturb
- Smart notification modes (sleep, gaming studying etc)
- Intelligent Notification filtering
- Advanced Notification filtering
- Smart Notifications (AI)
- Custom Notification Schedules
- Scheduled notification send
- Advanced notification schedules
- Interactive Notification previews
- Notification bookmarking
- Keyword/context based notifications
- Community Feedback Integration
- Priority contacts and channels
- Priority Inbox
- Cascading priority levels
- Productivity tools integration
- Enhanced Cross device syncing

- External configuration/keyword filtering

After placing all these ideas together, the team realized a lot of ideas fall under similar veins and decided to group a few of them together so that diverse design alternatives could be chosen.

4.2.3 *Selection of Design Alternatives*

The ideas chosen were decided to have three key features:

1. Content/Category Filtering Mechanism
2. Mode-Aware Filtering Mechanism
3. User-Defined Prioritization Mechanism

After voting and moving forward, three different design alternatives were chosen.

- Smart Modes (*Mode-Aware Filtering Mechanism*): This introduces a feature that adds customizable modes such as gaming, sleep, studying, etc. where users can preset their notification filtering and schedule depending on the activity.
- Advanced Notification filtering (*Content/Category Filtering Mechanism*): This allows users to filter notifications more precisely based on context, by one specific user/group of users or by specific keywords.
- Priority Contacts and Channels (*User-Defined Prioritization Mechanism*): This allows some channels or users to marked as more of a priority which causes their message to have a different visual/audio cue and allows them to show at the top of the inbox.

4.2.4 *Textual Prototypes*

- The **Smart Modes** feature introduces customizable modes to align notifications with the user's current activity, ensuring relevance and minimizing distractions. It allows for the grouping of notifications with the option to schedule or automatically adjust modes based on contextual clues.
- The **Content Filters** is an advanced notification filtering designed to empower Discord users with the ability to filter notifications based on specific content, categories, or user-defined criteria. This mechanism aims to ensure that users receive only the most relevant notifications, tailored to their interests, activities, and preferences.

- **PriorityHub** is user-defined prioritization feature that allows users to set their own priorities for notifications. Users can effortlessly mark certain contacts, channels, or conversation threads with different priority levels, which directly influence the visibility and order of notifications, ensuring that users see the most critical messages first.

5 INITIAL PROTOTYPING

Below we introduce the initial prototypes developed to enhance the Discord experience.

5.1 Prototype 1: Smart Modes

This prototype is designed to reduce the clutter of notifications for Discord users by introducing customizable modes like "Work," "Study," "Gaming," and "Sleep." These modes allow users to modify their notification settings to align with their ongoing activities, grounded in the idea that notifications should be relevant to the user's specific situation. Initially, all notifications are collected into a universal group to preserve traditional operation. Users have the option to assign notifications to specific groups and set rules at the group level, with the ability to include any notification source—be it direct messages, servers, or channels—into these groups. Group-specific preferences will override the general settings, and each group is designed to be exclusive, ensuring a notification source can only be assigned to one group.

This concept was inspired by feedback from our needfinding phase, particularly from a college student who missed essential academic notifications amidst numerous unnecessary alerts and a remote worker who faced issues with inadvertently muted channels. By enabling users to customize their notification settings with modes that are context-dependent, this feature is designed to transform interactions on Discord. It emphasizes important alerts based on the user's current activity, while minimizing irrelevant noise, significantly enhancing the overall experience.

5.1.1 *Core Components*

1. ***Scheduled Mode Selection/Automatic Mode Selection:*** Users have the option to set specific times for modes to activate or allow AI to automatically change modes based on contextual cues. For example, if it is 9

AM and there's an influx of work-related notifications, the system could deduce it's time to switch to Work Mode.

2. ***Backwards Compatibility:*** By keeping notifications in the default group, the system remains unchanged for users who prefer the existing setup.

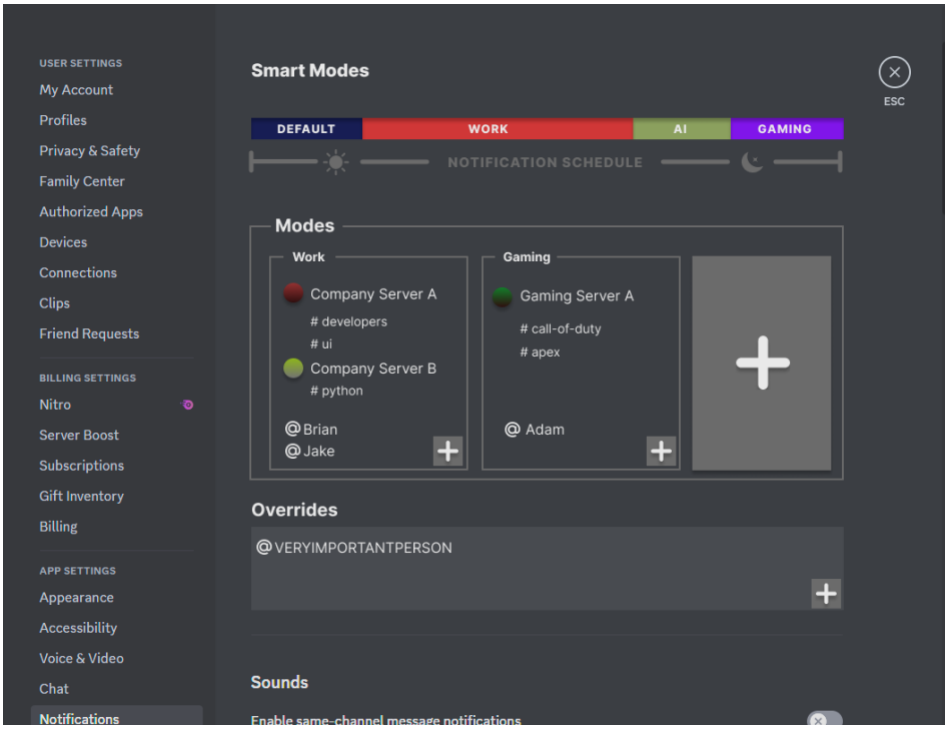


Figure 2—Medium fidelity prototype for new 'Smart Modes' within Notifications global settings.

5.1.2 *Design Details*

- ***Notification Modes Menu:*** The current notification settings would be relocated from "App Settings" to a standalone top-level setting. Within this area, a "Modes" menu would be introduced, facilitating the configuration and establishment of various modes.
- ***Mode Schedule:*** Within the Modes menu, users would have the ability to schedule specific modes, enabling these selected modes to temporarily supersede the default settings according to the user's preferences.
- ***Importance Override:*** The Modes menu will include an option for an importance override, which can be set through AI recommendations or defined explicitly by the user.

5.2 Prototype 2: Content Filters

This prototype is an advanced filtering mechanism designed to provide Discord users with the ability to tailor their notification stream by setting filters based on specific keywords, phrases, or predefined content categories. When a new notification arrives, the system checks it against active filters. If the notification matches any filters, it's either shown immediately, muted, or added to a digest, based on the user's preferences. The design of this prototype is a direct response to user feedback calling for greater control over the barrage of notifications experienced on Discord. By incorporating user control and freedom, the prototype allows users to effectively manage their cognitive load by setting preferences that reduce unwanted or less important notifications.

5.2.1 *Core Components*

1. ***Content and Category Labels:*** Users can assign labels to channels, messages, or specific content within Discord, categorizing them into easily identifiable groups.
2. ***User-Defined Filters:*** Users can create custom filters based on labels, keywords, or specific criteria, which are saved for future reference.
3. ***AI-Assisted Categorization:*** Use ML algorithms to analyze message content and automatically suggest categories, aiding in the filtering process.

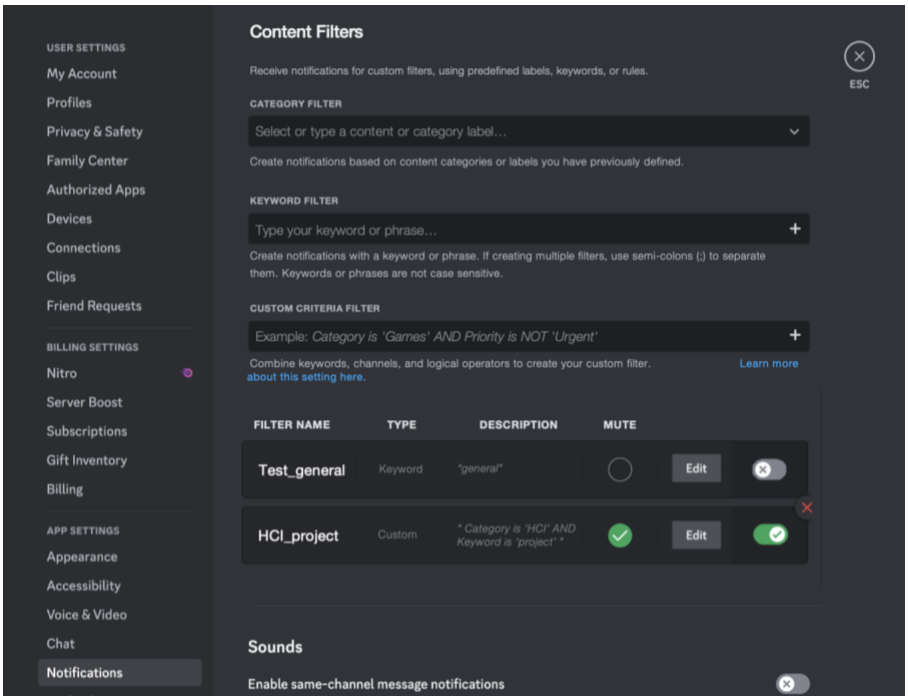


Figure 3—Medium fidelity prototype for new 'Content Filters' within Notifications global settings.

5.2.2 *Design Details*

- ***Filters section:*** A new section within Discord's global notification settings titled 'Content Filters' provides users with a dedicated space to manage their filters.
- ***Keyword Filtering:*** An input field labeled 'Keyword Filters' allows users to enter keywords or phrases to create new notifications.
- ***Category Dropdown:*** A dropdown menu titled 'Category Filters' offers predefined categories such as 'Game Updates', 'Direct Messages', 'Server Announcements', and more, for users to select as filters. The dropdown also includes a search functionality to quickly find specific categories.
- ***Custom Criteria Logic:*** An advanced option named 'Custom Criteria' opens a modal where users can create complex filtering rules using logical operators and conditions, much like setting rules in an email client. This modal includes examples and a 'Help' section to guide users through the process.

- ***Filters interaction options:*** Users can interact with the created filters by clicking on edit or delete options and toggle switches to quickly activate or deactivate specific filters ensuring ease of management.

5.3 Prototype 3: PriorityHub

PriorityHub distinguishes itself from Smart Modes and Content Filters through its allow-list approach, fundamentally altering the way notifications are managed within Discord. Unlike the aforementioned prototypes that aim to filter out the noise by identifying signals based on context or content, PriorityHub empowers users to proactively designate certain contacts or channels as high priority, ensuring that notifications from these sources are prominently displayed. This method offers users direct control over the notifications deemed important, significantly reducing the likelihood of missing crucial messages amidst a flood of alerts.

This concept was inspired by insights gathered during the needfinding phase, where a common user behavior emerged: many participants expressed a tendency to mute all channels by default and selectively enable notifications only for those few channels they deemed essential. Recognizing this, the PriorityHub concept aims to provide a more nuanced and user-friendly solution, allowing individuals to proactively identify and prioritize notifications from channels and contacts they genuinely care about.

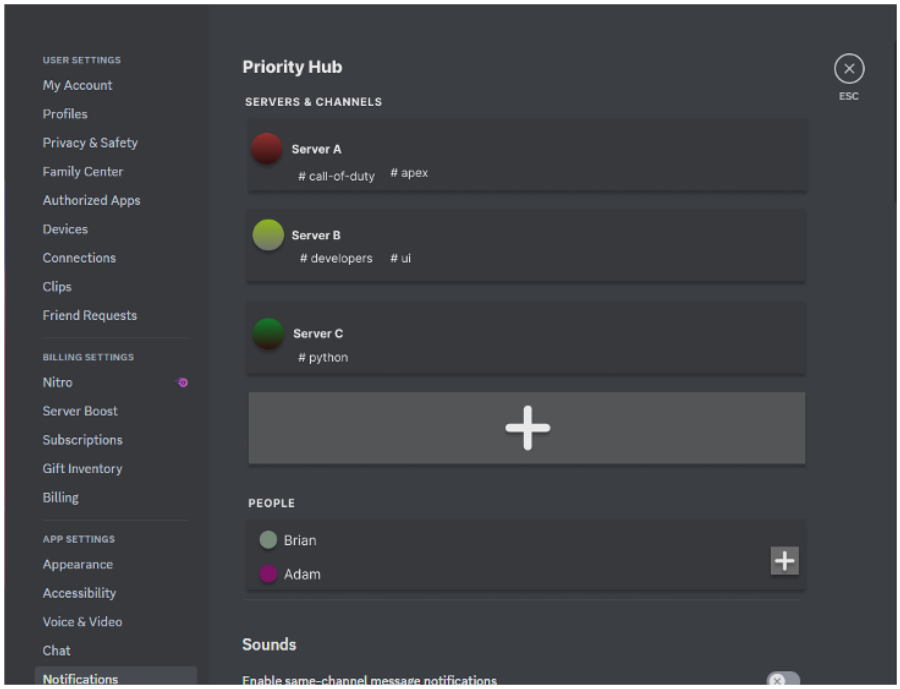


Figure 4—Medium fidelity prototype for new 'Priority Hub' within Notifications global settings.

5.3.1 *Design Details*

- ***Priority Settings Panel:*** A dedicated section within Discord's settings that allows users to manage their high-priority contacts and channels, including options to add, remove, or adjust priority levels.
- ***Notification Sorting:*** Automatically places high-priority notifications at the top of the notification center, ensuring they are the first seen.
- ***Visual Indicators:*** Utilizes clear visual indicators (e.g., stars, color coding) to highlight high-priority notifications, making them stand out from the rest.

6 INITIAL PROTOTYPES EVALUATION

6.1 Evaluation Plan

The purpose of this evaluation plan is to measure the usability, effectiveness, and user satisfaction of the three proposed Discord notification system prototypes: Smart Modes, Content Filters, and PriorityHub.

Our evaluation will involve a mix of participants from the Georgia Tech student community as well as Discord users who are not affiliated with the university.

This mix will ensure a diverse pool of participants, representing a range of use cases and perspectives. We will reach out to participants through Georgia Tech's internal communication channels, Discord community groups, and social media. The main goal is to determine which prototype best meets user needs for managing notifications within Discord's diverse user environment.

6.1.1 *Evaluation Methodology*

- ***Comparative Analysis:*** We will employ a structured survey approach, utilizing the PeerSurvey platform, to conduct a comparative analysis of three distinct prototypes. Participants will be provided with comprehensive visual materials and descriptive narratives for each prototype, ensuring they have a clear understanding of the functionalities despite not having direct interaction. The survey will incorporate a mix of Likert-scale questions for numerical ratings on aspects such as ease of use, perceived utility, and fit within participants' current Discord usage, along with open-ended questions to collect qualitative feedback. These questions will explore users' preferences, perceived efficiency of the prototypes, and their potential integration into existing Discord workflows. The structured nature of this survey, as detailed in *Appendix: Initial Prototypes Survey Questions*, ensures the collection of substantial data without necessitating direct prototype engagement.
- ***Predictive Analysis:*** To supplement the survey data, we will perform cognitive walkthroughs and heuristic evaluations. This predictive analysis foregoes direct user interaction, focusing instead on our team's systematic simulation of prototype usage to predict usability challenges and identify potential user difficulties. The findings will help us anticipate where the user struggles and inform necessary design refinements. Both cognitive walkthroughs and heuristic evaluations will serve as supplementary methods to the primary data gathered through user surveys. They will provide additional insight into the user experience with the prototypes, helping to triangulate the data collected from the surveys.

6.1.2 *Metrics & Analysis*

Our comprehensive analysis will integrate **quantitative metrics**—like feature usage frequency, efficiency, feature prioritization, and preference ratings—with

qualitative feedback— user opinions on usefulness, ease of use, and overall experience. Quantitative data will undergo statistical analysis, while qualitative feedback will be thematically categorized to identify prototype strengths and improvement areas.

This multidimensional evaluation will create a detailed picture of user reception, potential impact, and suggestions for refining the features under consideration. The survey-based method will provide early insights into user expectations and acceptance, guiding the final design choices to ensure the selected features or prototype combinations effectively address user needs in managing Discord notifications. Predictive evaluations, including cognitive walkthroughs and heuristic analyses, will further enrich our analysis by identifying usability issues early on, allowing for targeted design improvements. This integrated approach ensures our final design decisions are well-informed and user-centered, ultimately enhancing user satisfaction.

6.2 Evaluation Results

6.2.1 *Comparative Analysis*

Overall, the survey results were greatly helpful in filtering down the prototypes we want to move forward with. The overall count of participants was n = 17 with nearly 100% of the candidates being test run all the prototypes.

Smart Modes had an overall positive interaction with 100% of participants finding the feature useful in managing notifications. The qualitative feedback was greatly useful with a few notable pieces of feedback:

- Too much information given at once
- Clumsy/slow to set up if you're part of many servers
- Additional schedules for different days of the week

When looking at a few more rating data we gathered," identifying and switching between different smart modes was straightforward" received a rating of 4.23 and "setting up and customizing smart modes felt intuitive and hassle free" received a score of 4.05 garnering generally positive interactions.

Content Filters overall had positive reviews but did the worst out of the three prototypes. Three participants did not find the interface as valuable in managing notifications. The rating questions also did slightly worse than the other two

prototypes. “the feature to create and apply content filters was clearly presented in the settings” got a rating of 4 while “configuring my filters and understanding their impact on my notifications was intuitive” got a rating of 3.7 (the worst of all the rating questions). Considerable qualitative feedback was also given:

- Too cluttered
- Notifications flags might be better
- Using AI may have some negative implications
- Might be too complex

The last prototype that we evaluated was PriorityHub. Overall priority hub received largely positive feedback with an approval rating of 100%. Furthermore, “the visual cues for high priority notifications were clear and helped in recognizing important alerts” received a score of 4.29 and “managing priority contacts and channels within in priority hub was a straightforward process” received a score of 4.35. Some feedback for this included:

- Might be too complex
- Might need to make the priorities more clear
- Group together one channels as a priority

When comparing the prototypes with each other, the two best candidates seemed to be Smart Modes and Priority Hub. The approval rates for both were 100%.

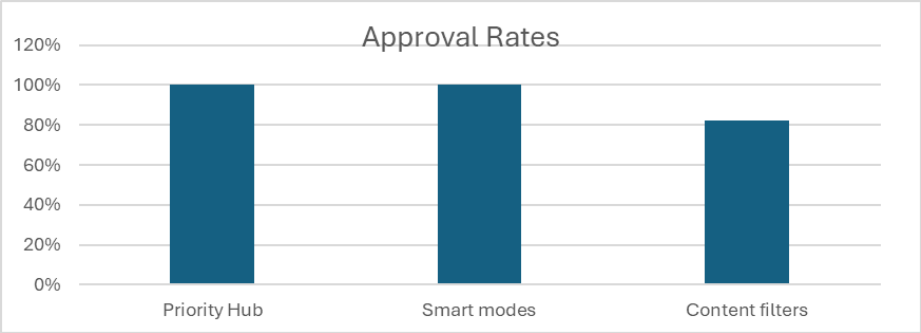


Figure 5 — Approval rates comparison between Smart Modes, Content Filters, and PriorityHub prototypes.

Furthermore, when users had to select one prototype. The results were overwhelmingly sided with Smart modes and Priority Hub with Priority Hub having a slight edge.

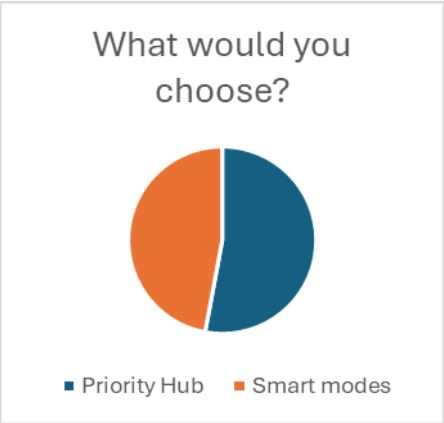


Figure 6—Pie chart for users' most preferred prototype.

Overall, after team discussions and analysis, we decided to move forward with priority hub as it seems to be evaluated well and would be much more versatile and usable as a high-fidelity prototype.

6.2.2 *Predictive Analysis*

In our predictive analysis, we employed various strategies including heuristic evaluations and cognitive walkthroughs. The central premise behind predictive analysis is that it does not fully represent the complete scenario; rather, it serves to augment the direct feedback received from users. Despite the inherent biases, the exercise of empathizing with our users and critically examining the challenges posed by our prototypes is invaluable. As frequent users of Discord, our familiarity with the platform enhances our participatory role in the design process, making it a prime example of action research.

During the cognitive walkthrough, we assessed how users might interact with each prototype by simulating specific tasks.

- For the **Smart Modes** prototype, the tasks were: 1) Adding modes, and 2) Adding overrides. We noted that adding modes was relatively straightforward due to the clear visibility of current modes and an obvious

"add" button, which serves as a visual cue for users to introduce new modes. However, adding overrides proved more challenging. The presence of both a text box for typing overrides and an addition button could potentially confuse users. Additionally, the layout of the Smart Modes was somewhat perplexing, with no clear indication of when each mode would activate.

- For the **Content Filters** prototype, the tasks included: 1) Adding a category filter, 2) Adding a keyword filter, 3) Implementing a custom criteria filter, and 4) Modifying existing filters. Our walkthrough revealed that while typing in a content filter was intuitive, it was not always clear which section was designated for input. The presence of a mute button and a toggle for enabling or disabling the filter also introduced elements of confusion.
- The tasks for the **Priority Hub** prototype were straightforward: Adding servers, channels, and individuals to prioritize. This process was unproblematic, but the prototype lacked features for editing or removing items from the Priority Hub, which could limit user control and flexibility.

In our heuristic evaluation, we selected three of Nielsen's 10 Usability Heuristics: 1) Visibility of System Status, 2) Consistency and Standards, and 3) Error Diagnosis and Recovery.

- **Smart Modes** demonstrated good visibility of system status by displaying different modes applicable to various days, although it failed to specify the exact timings for each mode. The override functionality was found to be inconsistent with standard practices and appeared novel, potentially leading to user confusion. Moreover, the absence of options to delete or edit modes was a significant design oversight.
- **Content Filters** displayed adequate visibility of existing filters, though the design diverged from traditional layouts by offering three separate input boxes for filters. The prototype facilitated error recovery well, with straightforward options to edit and deactivate filters.
- **Priority Hub** effectively showcased all prioritized hubs, servers, channels, and individuals with high visibility and consistent design throughout. However, the lack of delete functionality could hinder users from correcting mistakes easily.

Overall, our cognitive walkthroughs and heuristic evaluations uncovered several usability challenges across the prototypes. Despite these issues, the simplicity and overall user-friendly design of the Priority Hub were commendable, with it demonstrating the fewest complications among the prototypes tested.

6.3 Raw Results

Raw data containing participants' responses to the survey are included in *Appendix: Initial Prototypes Survey Responses* for reference.

7 SECOND ITERATION PLAN

7.1 Planning Overview

In our initial effort to refine Discord's notification system, three prototypes were developed: Smart Modes, Content Filters, and PriorityHub. Based on thorough comparative and predictive analyses, PriorityHub emerged as the superior prototype in terms of user satisfaction and usability. As we transition into the second iteration, our focus will be on enhancing PriorityHub by directly addressing the feedback and gaps identified in the first round.

This phase aims to solidify PriorityHub as a robust solution that effectively balances intuitive design with effective management of user controls and preferences. The decision to further develop PriorityHub was driven by its clear advantage in handling notifications more intuitively compared to the other prototypes.

7.2 First Iteration Insights

From the survey and predictive analysis, PriorityHub was identified as a strong solution for managing notifications, highlighted its clarity and ease of use. However, feedback also stressed areas for improvement:

- **Complexity and Clarity**: Users found some aspects of PriorityHub complex, suggesting a need for simplifying the interface to enhance usability without sacrificing functionality, particularly in prioritizing notifications and managing them.
- **Feature Limitations**: The absence of options for editing or removing items was noted, pointing towards a need for increased user control and flexibility.

The insights from the initial evaluation provide a solid foundation for the continued development of PriorityHub. Therefore, a new dedicated needfinding activity is not necessary at this stage. Instead, our efforts will focus on addressing the specific improvements identified.

7.2.1 *Questions and Areas for exploration*

The primary gap identified is the interface complexity and the need for additional customization options that allow users to set preferences more precisely. Several questions and areas for exploration have been identified to guide the next phase of development:

1. **Interface Simplification**: How can we simplify the user interface of PriorityHub while maintaining its comprehensive functionality?
2. **Feature Expansion**: What additional features can be integrated to address user demands for more detailed control without overwhelming the user?
3. **AI Integration:** Can artificial intelligence be employed to dynamically adjust notification priorities based on changing user patterns and behaviors?

7.3 Design Considerations in Next Iteration

To refine PriorityHub into a more effective tool, we will focus on the following design considerations:

1. **Enhanced User Control**: Introducing functionality to edit and delete priorities and setting priorities at different levels. This will allow users to adapt the system to their changing needs.
2. **Clear Visual Cues**: We will develop and integrate improved visual cues such as color coding and iconography to help users quickly recognize important notifications.
3. **Seamless Integration**: Redesigning the priority settings panel to maintain a consistent interface design, ensuring that Priority Hub integrates seamlessly with Discord's existing notification framework.

7.4 Next Phase Goals

The primary focus of the next phase is to directly address the feedback from the initial evaluations, with a clear aim to increase usability and satisfaction:

- **Enhancing Usability**: We will ensure that PriorityHub's features are easy to use and understand, minimizing the need for extensive user education and increasing intuitive interaction.
- **Expanding Functionality**: We will develop new features to allow users more granular control over their notification preferences, making PriorityHub adaptable to varied user preferences and lifestyles.

By concentrating on these goals, the second iteration aims to not only refine PriorityHub but also to ensure it aligns perfectly with the users' daily needs on Discord, thereby enhancing their overall experience on the platform. Each goal will be measured against specific user feedback and satisfaction metrics to ensure the enhancements meet the expected usability improvements.

8 FINAL PROTOYPE

To develop our high-fidelity prototype, we selected the collaborative design tool, Figma. Below, we have shared some key screenshots for an overview. However, the complete prototype is directly accessible here.

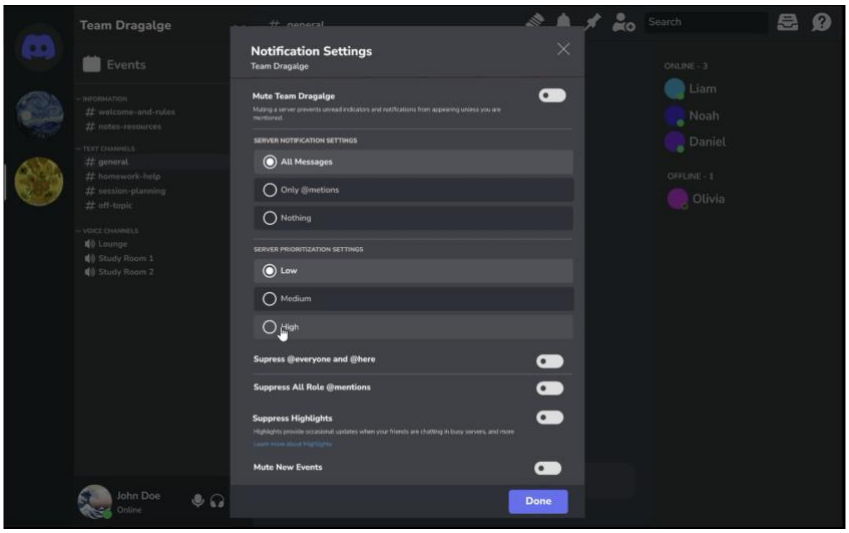


Figure 7— Prioritize a server.

In developing the high-fidelity prototype, our goal was to effectively capture the primary user flow for prioritization. This includes several key actions: prioritizing a server, channel, and direct message (DM); checking an inbox for prioritized messages; viewing prioritization settings; and toggling visual indicators on and off. While there are additional features that complement these core functions, the ones listed encapsulate the fundamental aspects of the feature.

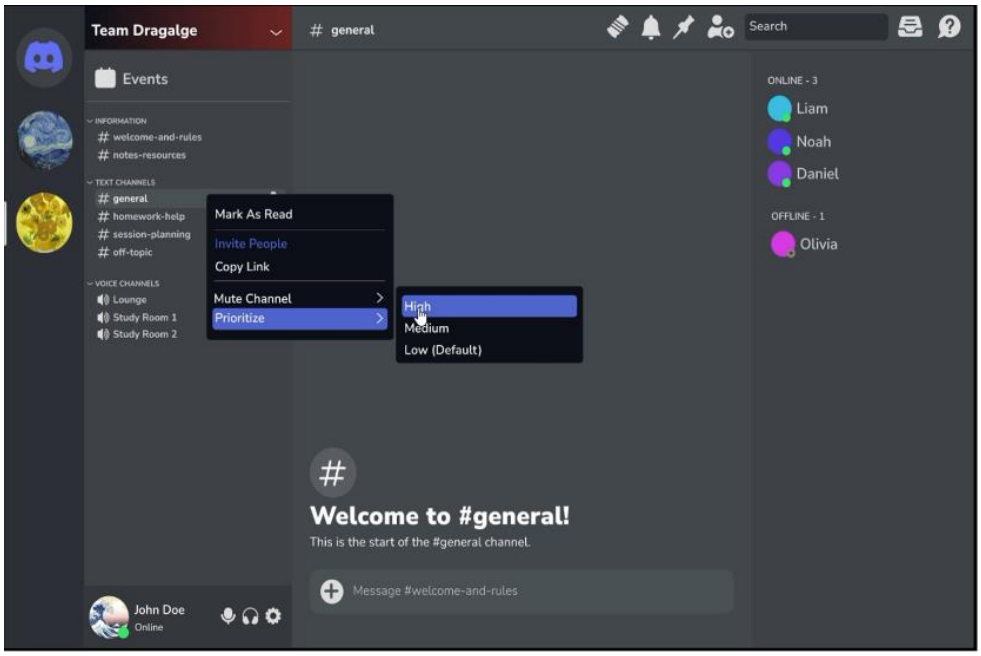


Figure 8—Prioritize a channel.

User feedback on the initial prototypes revealed a preference for a simpler, more intuitive interface that emphasized clear prioritization. With this feedback in mind, we refined the initial prototype while preserving its most appreciated features, such as visual cues and straightforwardness. To streamline the experience, narrowed the scope of prioritization to just three distinct levels: low, medium, and high. This simplification not only reduces the complexity of choices available to users but also makes the implications of each priority level immediately understandable. This also lets us add simplicity to the design while maintaining the level of complexity that the majority of users wanted. By adding this constraint to the interface, we effectively guide users in their decision-making process, ensuring that the system is both manageable and easy to navigate. Additionally, we established a default setting that assigns a low

priority to all notifications. This strategy ensures seamless integration and backward compatibility with the existing Discord notification system, guaranteeing a smooth transition for users. Those who opt not to use the new prioritization features will experience no change in their notification functionality, allowing them to continue as usual without any disruptions or necessary adjustments. This approach exemplifies the principle of flexibility from universal design by offering multiple ways of using the system. Users can either engage with the prioritization system to tailor their experience according to their specific needs and preferences or interact with the unchanged, standard interface.

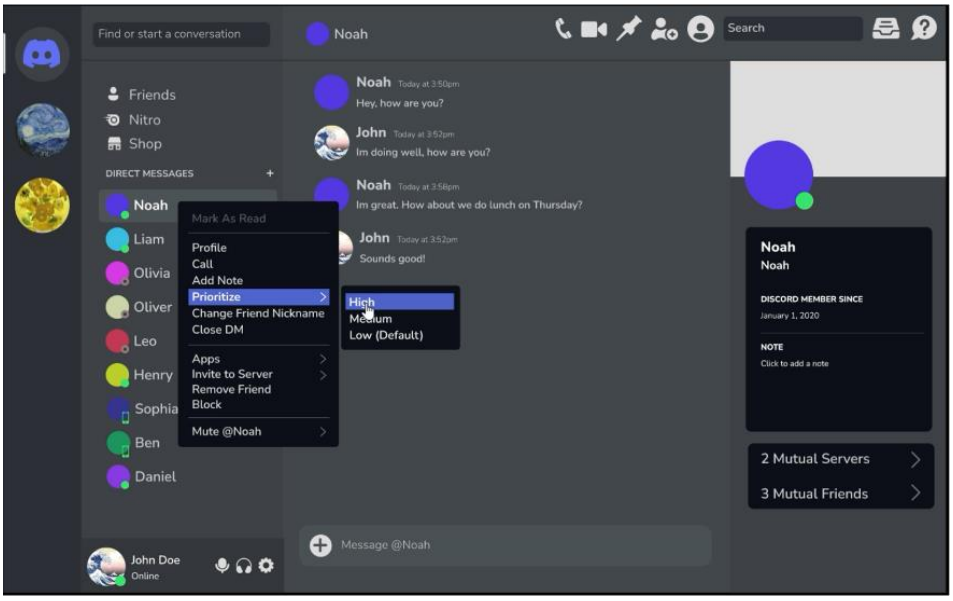


Figure 9—Prioritize a DM.

Conversely, for users who decide to engage with the prioritization features, the system offers a structured approach to managing notifications. After setting priorities for different types of communications, notifications are displayed first by priority level and then in chronological order. This method facilitates a quick and easy way for users to discern their notifications, organizing them in a sequence that places the most important items at the forefront. This also lets the user utilize mapping to intuitively understand how the priority levels affect the order of servers/users.

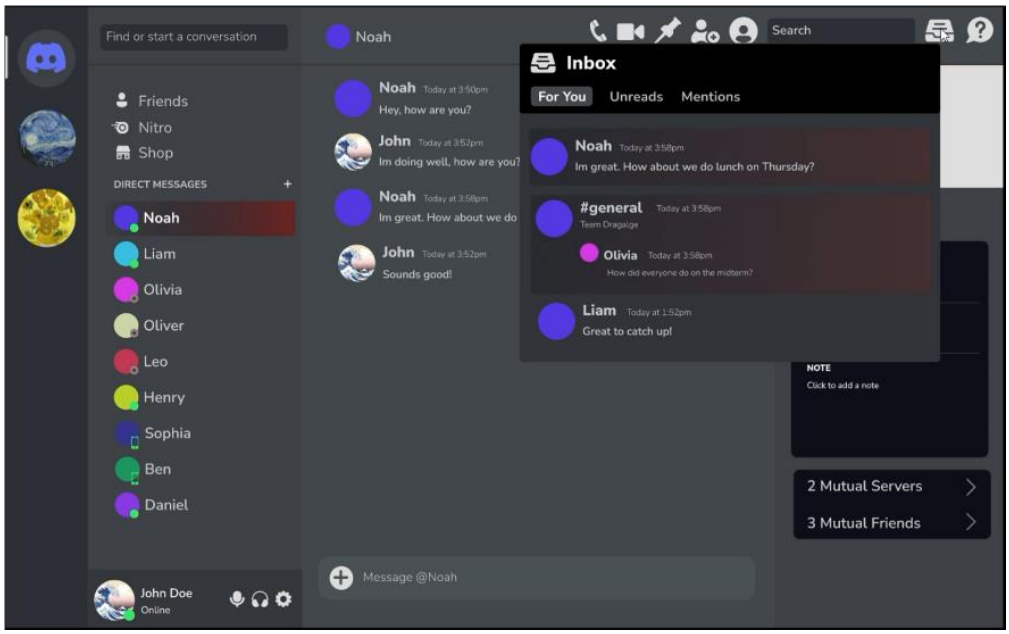


Figure 10—View notifications from prioritized sources highlighted in inbox.

One thing to note is that on the prioritization system for servers and channels, there is a hierarchical parent-child relationship where the server is the parent, and its channels are the children. Within this structure, the priority assigned to a parent (server) does not override the priority of a child (channel) unless the parent's priority is higher. For example, if a server is assigned a high priority and a channel within it is assigned a low priority, messages from that channel will still be displayed with high priority. Thus, setting a server to high priority effectively applies the same high priority level to all its channels. It's important to note that this parent-child relationship does not exist for direct messages (DMs), so there are no similar concerns or complications with prioritization in DMs.

The interface enhances user awareness of message prioritization through visual signifiers, applied to various types of communications both inside and outside the inbox. It employs a subtle color scheme to distinguish between priority levels: communications marked as medium priority are highlighted in yellow, and those deemed high priority are highlighted in red, facilitating immediate recognition. The default setting for low priority retains the original color scheme, preserving the application's look and feel after this feature is added.

Additionally, the settings offer an option for users to remove all visual indicators of prioritization.

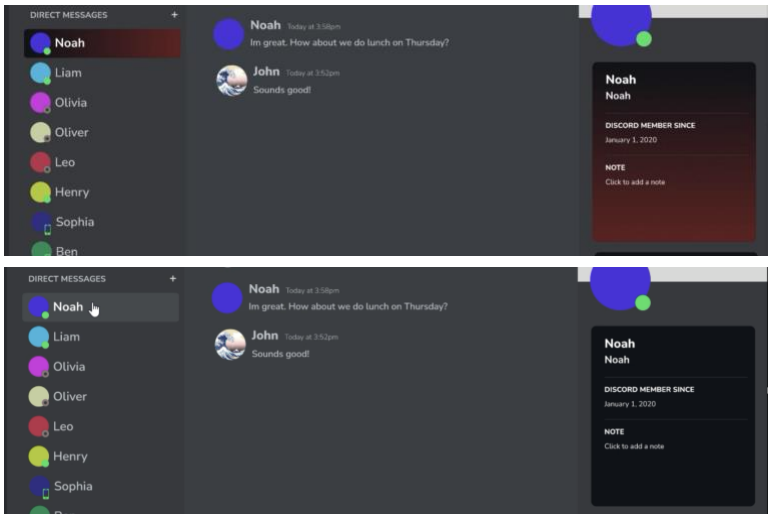


Figure 11 — Visual prioritization indicators for contacts enabled (top) and disabled (bottom).

This flexibility allows users who prefer a less cluttered interface to disable these cues, ensuring the system can be tailored to meet diverse needs and preferences, enhancing comfort and usability for those who do not require visual prioritization feedback.

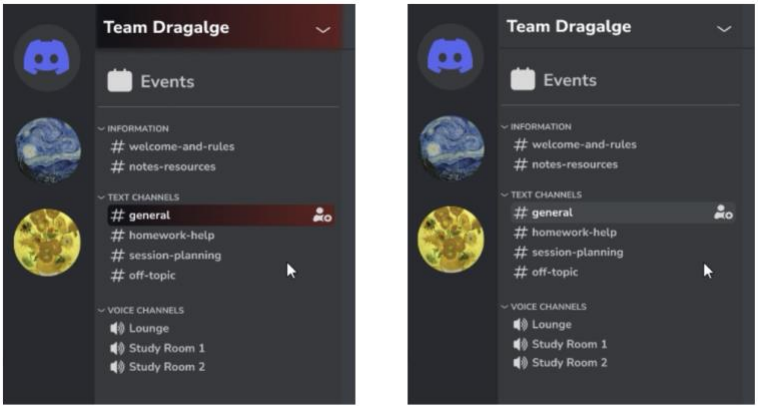


Figure 12 — Visual prioritization indicators for channel and server enabled (left) and disabled (right).

Building on the theme of customization, the settings screen in the application plays a crucial role in allowing users to further personalize their experience.

Here, users can not only toggle visual indicators for prioritization but also see all channels, servers, and DMs they have set priorities for, allowing for easy review and management.

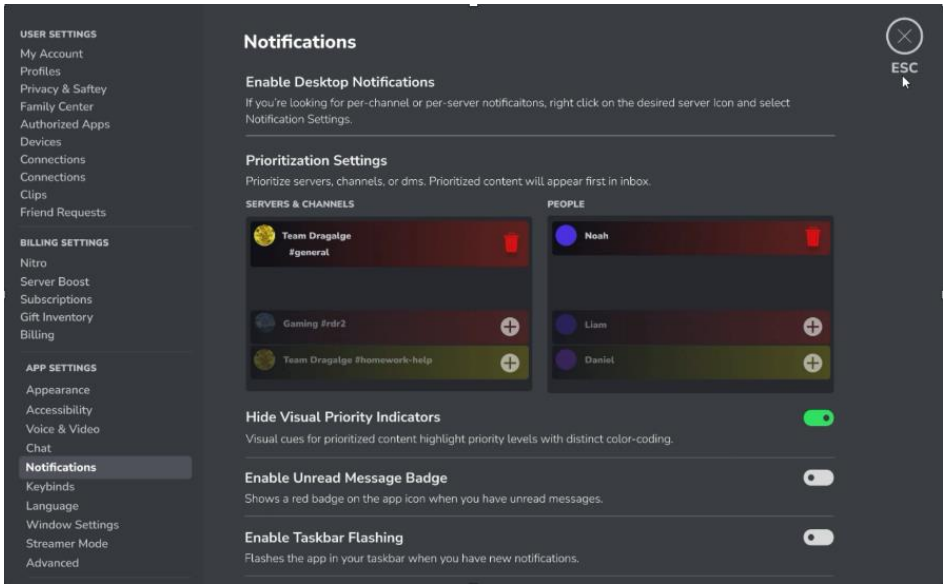


Figure 13 — Prioritization Settings panel within global Notifications settings with option to hide visual indicators enabled.

The settings incorporate a "quick add" feature to add prioritizations, intelligently suggesting the most appropriate priority levels based on the user's interaction patterns within Discord. This feature might recommend a higher priority for frequently used DMs or active servers. This streamlined approach not only tailors settings to individual usage patterns but also significantly reduces cognitive load, making the management of notifications straightforward and user friendly.

We intentionally designed the settings screen to integrate seamlessly into the existing Discord notification framework, rather than creating a separate interface. Our goal was to maintain a cohesive user experience by ensuring that the new features felt like a natural extension of the current system. This integration strategy minimizes disruption and learning curve for users, as they can manage their notification settings without having to navigate away to a different part of the application. By embedding our settings within the familiar context of Discord's existing notification management area, we enhance user

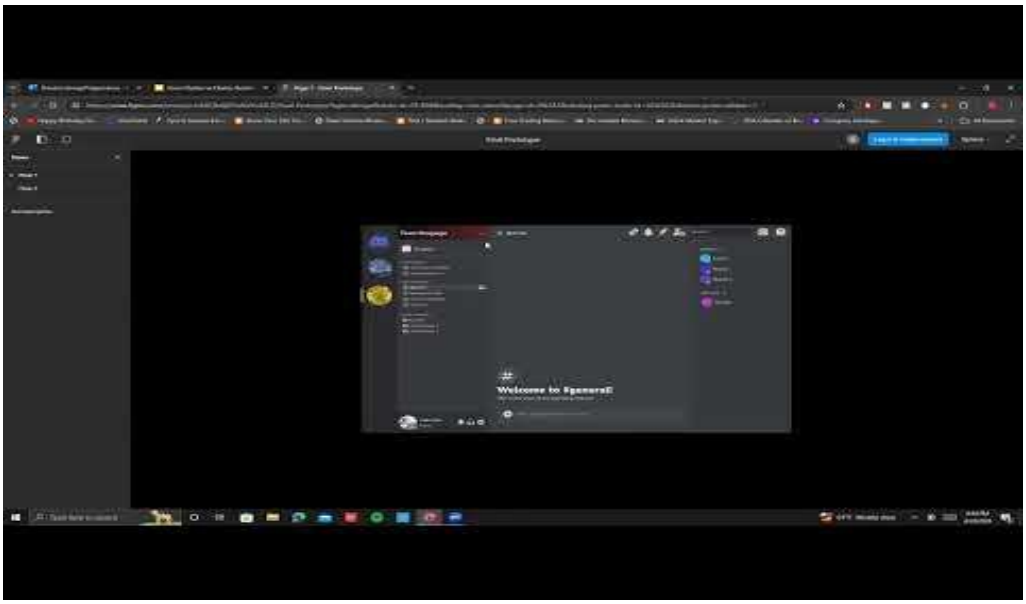
comfort and streamline the adoption process, making it easier for users to utilize and benefit from the new prioritization features.

In conclusion, our high-fidelity prototype illustrates a comprehensive integration of prioritization features within the existing Discord environment, aimed at enhancing user experience through intuitive design and personalization. By incorporating user feedback and adhering to universal design principles, we have crafted a system that not only simplifies the management of notifications but also provides users with the flexibility to adapt the interface to their specific needs. The integration of the settings screen ensures that users can easily access and customize prioritization settings without feeling overwhelmed, thereby maintaining the application's usability and appeal. This balance of functionality and user-centric design demonstrates our commitment to creating an accessible and efficient platform that respects user preferences and improves their overall experience on Discord.

9 VIDEO PROTOTYPE

The video prototype provides a walkthrough of the new PriorityHub prototype. The video prototype is linked below:

https://youtu.be/wkXOMXNmnCk



10 FINAL EVALUATION PLAN

Our comprehensive evaluation plan builds upon the foundational elements outlined in our initial evaluation plan.

10.1.1 *Evaluation Methodology*

We will employ both comparative and predictive analysis methodologies to assess our final prototype.

- **Survey Analysis:** We will develop and distribute a detailed survey to a targeted group of participants, specifically current Georgia Tech students enrolled in our class. The survey will feature fifteen diverse questions designed to gauge the usability, effectiveness, and overall satisfaction with our final prototype. Questions will focus on whether the prototype meets expected outcomes, how it compares to pre-established criteria, and whether it fulfills the intended scope. Additionally, we will solicit recommendations, comments, and any pertinent feedback that could provide insight into whether our developmental trajectory aligns with user needs and expectations.
- **Predictive Analysis:** This will involve cognitive walkthroughs and heuristic evaluations, aimed at understanding potential user experiences with the final prototype. These evaluations will help us anticipate user interactions and identify any possible challenges or obstacles that may arise during use.

10.1.2 *Metrics & Analysis*

Our evaluation will incorporate both qualitative and quantitative feedback. Qualitative insights will primarily be derived from open-ended survey responses and the outcomes of our predictive analysis, which will challenge our assumptions and enhance our understanding of user experience dynamics. Quantitative data will be collected through survey responses that quantify user opinions and experiences into measurable data, facilitating statistical analysis and visualization.

The culmination of this evaluation will provide a clear depiction of how effectively our design has enhanced the user experience in managing

notifications from the Discord application. By combining qualitative observations with quantitative data, we aim to refine our prototype into a more user-centric solution, ensuring it not only meets but exceeds user expectations.

Through this dual approach, we intend to gather comprehensive insights into the overall impact and effectiveness of our design within the user community, enabling us to make informed decisions on future enhancements and iterations.

11 FINAL EVALUATION RESULTS

11.1 Survey Analysis

This analysis is comprised of responses from 20 participants, all recruited from Georgia Tech student community. Leveraging the PeerSurvey platform for data collection, we sought to capture diverse perspectives that would inform our developmental trajectory and ensure alignment with user needs and expectations. For further reference, the raw data containing participants' responses to the survey are included in *Appendix: Final Prototypes Survey Responses*.

Our findings, both quantitative and qualitative, provide insight on users' perceptions of usability, functionality, effectiveness, and areas for improvement for the PriorityHub prototype.

11.1.1 *Quantitative Findings:*

For the statement related to improving how notifications are managed (Q2), the average rating was 4.15, predominantly within the 'Agree' category. This suggests that users perceive these features as beneficial enhancements to their notification management experience.

Similarly, when asked about the direct control over important notifications (Q3) the responses were even more favorable, with an average rating of 4.45. A significant portion of users 'Strongly Agreed' that the features would provide enhanced control, signifying strong support for these changes.

Feedback on the features that provide a nuanced and user-friendly approach to prioritizing notifications (Q4) was also positive, with an average rating of 4.05. Although most feedback was favorable, ranging from 'Agree' to 'Strongly Agree', there were a few 'Neutral' and 'Disagree' responses, suggesting that while most

users are in favor, there is a small group that believes that the features could benefit from further adjustments.

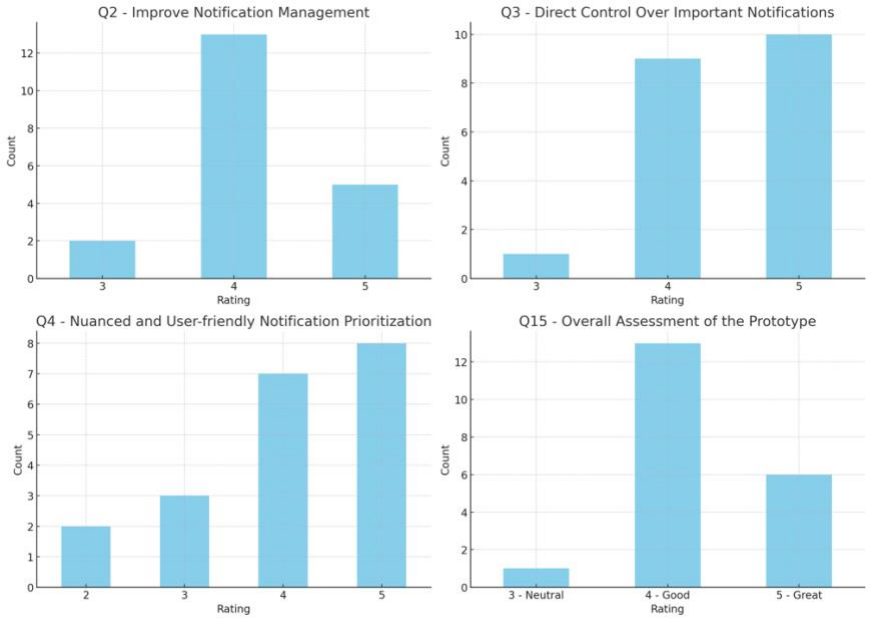


Figure 14 — Distribution of responses for feedback on the proposed notification management features in Discord.

Overall, when assessing the final prototype (Q15), it was mainly rated as 'Good', with several 'Great' responses. This demonstrates that users generally view the prototype as a valuable improvement.

11.1.2 *Qualitative Findings:*

Participants generally expressed approval for the proposed features' potential to improve *usability and functionality*, appreciating their practical applications in effectively managing notifications. These features were seen as enhancements over Discord's current interface, particularly in scenarios involving high message volumes. Regarding *effectiveness*, users noted the features' capability to better control notification prioritization, comparing them favorably to similar functionalities in platforms like Slack. Although some feedback suggested these features were not completely novel, their effectiveness was viewed positively. Lastly, the *area of improvements* was less emphasized in the responses, with most participants expressing satisfaction with the current state of the prototype and pointing out only minor areas for refinement.

Overall, the sentiment towards the prototype was positive. Respondents recognized the improvements in user experience, noting the features as beneficial and supportive. While some minor areas for improvement were noted, the general direction and functionality of the prototype received strong approval, emphasizing its potential to enhance user engagement on Discord.

11.2 Predictive Analysis

For our final prototype predictive analysis, we employed a cognitive walkthrough and a heuristic evaluation.

During our cognitive walkthrough, we evaluated potential user interactions with the final prototype by simulating specific tasks designed to explore the system's usability. The tasks involved changing the priority settings of a server, a channel, and an individual, as well as reviewing messages in a prioritized inbox and adjusting notification priority settings. Our findings indicated that the processes for adjusting the priority of a server, channel, and person were all straightforward and intuitive. Similarly, the functionality of displaying priority messages at the top of the inbox was also clear and effective. However, we encountered some issues with the intuitiveness of the notification settings; the separation between the settings for servers and channels versus those for individuals seemed unnecessary and confusing, suggesting a lack of clear rationale for this distinction.

In our heuristic evaluation, we focused on three of Nielsen's 10 Usability Heuristics: Visibility of System Status, Consistency and Standards, and Error Diagnosis and Recovery. The prototype excelled in providing clear visibility of the various priority levels assigned to servers, channels, and individuals, ensuring users could easily understand and interact with these settings. Despite this, we noted an inconsistency in the separation of priority settings for servers and channels versus individuals, which did not align with our standards for a cohesive user experience. Regarding error diagnosis and recovery, the prototype performed well; the prioritization inbox and highly visible, user-friendly notification settings facilitated effective management of errors and adjustments.

Overall, while the prototype has room for improvement, it demonstrated significant strengths in both the cognitive walkthrough and heuristic evaluation.

This thorough testing highlighted the prototype's effective design and usability, setting a strong foundation for further refinement.

12 INDIVIDUAL REFLECTIONS - NELLIE CORDOVA

12.1 Individual Contributions

I contributed to project ideation, drafting the introduction, creating survey questions and heuristic evaluations. I conducted user interviews, played a part in creating the Content Filters prototype and providing design input for the final prototype. I also set up evaluation plans for different phases (needfinding, brainstorming and second iteration) and completed the final survey analysis.

12.2 Team Member Contributions

All team members made important contributions to the project. Brandon showed responsiveness and initiative throughout, he not only conducted user interviews but also designed multiple prototypes (2 initial prototypes and the final version) and provided detailed descriptions for each. He also assisted with the evaluation plan and conducting survey analysis for the needfinding phase. His dedication and proactive approach greatly enriched our project. Luis completed tasks assigned to him, such as the user interview analysis for the needfinding phase, final evaluation planning including creating survey questions and conducting the predictive analysis. Roshen actively participated in user interviews, contributed to the brainstorming phase, and completed tasks such as survey analysis for initial prototypes, design justification and video creation for the final prototype.

12.3 Overall Reflections

Overall, our project had its strengths and areas for improvement. Our collaborative brainstorming session is a perfect example of how synchronic communication can enhance teamwork and productivity. However, we faced challenges with maintaining momentum and meeting deadlines, partly due to the asynchronous nature of our work. While I took the lead in communication and organization, better communication and task division were essential for progress. I believe tools like JIRA could have helped us stay on track. Moving forward, I'm confident we can apply lessons learned to future projects. Lastly, I want to thank the team for their efforts in working on the redesign of Discord interface. Throughout the project and course, my understanding and appreciation for user-centric interfaces have grown significantly.

13 INDIVIDUAL REFLECTIONS - LUIS OCHOA

13.1 Individual Contributions

My individual contributions consisted of reflecting on the user interviews, predictive analysis of both the final and initial prototypes, writing the final prototype survey questions, and creating the evaluation plan to test our final prototype.

13.2 Team member contributions

Overall, everyone on the team contributed a very large amount of work. Brandon contributed to the need-finding, brainstorming, evaluation of the initial need-finding survey and the creation of many of the prototypes. Nellie contributed to the introductions, brainstorming, heuristic evaluations, the initial prototype, and the final evaluation results. Roshen contributed to the brainstorming, video prototype, survey evaluations, and the final prototype justification. Everyone contributed to editing the document.

13.3 Overall reflections

Overall, I am happy to have worked alongside the other team members on this project. I believe everyone held their own and contributed such tangible and impressive sections of this group project. We all did an amazing job at brainstorming together, communicating with each other, assigning internal deadlines, and taking moments to reflect on the process of improving an aspect of the Discord application.

14 INDIVIDUAL REFLECTIONS - ROSHEN JAGAJEEVAN

14.1 Individual Contributions

My individual contributions to the project mostly dealt with brainstorming, the initial prototype survey evaluations, parts of the final prototype justification and the video prototype.

14.2 Team member contributions

Overall, my different team members each contributed a very large amount of work. Nellie contributed to the initial project starting, introductions, large parts of brainstorming, heuristic evaluation, the initial prototype, and the final evaluation results. Brandon contributed greatly to need finding, brainstorming, evaluation of the initial need finding survey and creation of majority of the prototypes (2 of the initial and the final prototype). Luis contributed greatly to the brainstorming, evaluation planning and predictive evaluation. Special thanks to Brandon and Nellie as they thoroughly looked through the document and worked heavily on formatting and really bringing the whole story together.

14.2.1 *Overall Reflections*

Overall, I was very impressed throughout the course of the project. The planning and progress went relatively smoothly. I did feel as if I got off to a little bit of a slow start in the beginning, but different team members took on a lot of leadership responsibilities such as Nellie with organizing tasks for the team and Brandon with taking initiative with the prototypes on Figma. The asynchronous setting, although it was a different experience, still went smoothly as any communication on the timing of work or when various things would be completed were easily updated in the WhatsApp group chat. Overall, I felt my team was able to communicate very well and I had a lot of enjoyment working on the project as I felt we were working through a redesign that could have an impact on discord.

15 INDIVIDUAL REFLECTIONS - BRANDON WATTS

15.1 Individual Contributions

My contributions to the project include project ideation, conducting user interviews for needfinding, evaluation of the initial needfinding survey, brainstorming, creation of 2 of the 3 initial prototypes (Smart Modes & PriorityHub), and creation of the final prototype in Figma.

15.1.1 *Team Member Contributions*

My team members were extremely helpful and crucial to the project's success. Nellie contributed to project ideation, writing the introduction, creation of the initial needfinding plan, creating the initial needfinding survey, heuristic evaluation, brainstorming, creation of an initial prototype, and final evaluation results. Roshen contributed to evaluation results, final prototype, the video prototype, and user interviews. Luis contributed to evaluation planning, final evaluation planning, brainstorming, video prototype, and survey evaluations.

15.1.2 *Overall Reflections*

Overall, the project went smoothly. Nellie was key in keeping everyone on track and distributing tasks effectively. Despite initial concerns about asynchronous communication leading to confusion, I was surprised at how well everyone's efforts came together. We made good use of our tools, collaborating on documents, leaving inline comments, and using WhatsApp for updates and questions.

One improvement could have been setting clear roles from the start. However, the Kanban approach was flexible and adapted well to varying availability among team members. Some weeks I could devote a lot of time to the project, while other times personal commitments limited my involvement, but others were able to step in. This flexibility helped, though it was sometimes hard to tell when team members were struggling due to our asynchronous communication.

Additionally, I wish I had a better grasp of the collaboration tools we used, particularly Microsoft Whiteboard and Figma. It was my first experience with both platforms, and I feel I didn't make the most of their capabilities. A deeper

understanding of these tools from the start would have likely enhanced my contributions.

16 REFERENCES

1. Discord. (n.d.). Notifications Settings 101. Retrieved from https://support.discord.com/hc/en-us/articles/215253258-Notifications-Settings-101
2. Google LLC. (2022, March 30). Discord Server Notification Settings [Video]. YouTube. https://youtu.be/o9OtHiqvrNw?si=_CIBo55SmKaLXhUO
3. Norman, D. A. (2013). The design of everyday things. MIT Press.
4. 10 Usability Heuristics for User Interface Design. (2024, January 30). Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/
5. Pesavento, R., Mandliya, R., & Akbar, I. (2024, February 4). Using Machine Learning to Build a Delightful Notification Experience. Discord. https://discord.com/blog/building-delightful-notifications-using-ml

17 APPENDICES

17.1 Appendix: Needfinding User Interview Questions

Participant Information

1. Age Range: *[Under 18, 18-24, 25-34, 35-44, 45-54, 55-64, 65+]*
2. What best describes your primary use of Discord? Please check all that apply. *[Gaming, Education, Work/Professional, Socializing, Other (Please specify)]*
3. If 'Other' was selected in the previous question, please specify: [*Open-ended*]
4. How often do you use Discord? *[Daily, Several times a week, Weekly, Less frequently]*

Interview Questions

5. What is your overall experience with Discord notifications like?
6. How do you currently manage your notifications on Discord? (e.g., muting channels, custom settings)
7. Have you experienced any issues with notification settings or customization?
8. Thinking about your ideal notification system on Discord, what features or changes would you like to see implemented?
9. Are there specific types of notifications (e.g., direct messages, server alerts) that you prioritize over others? How do you decide?
10. Can you share a scenario where you missed an important notification on Discord? What was the impact, and how could it have been avoided?
11. Conversely, have you ever been overwhelmed by notifications on Discord? Describe that experience and how you dealt with it?
12. In terms of customizing notifications, what level of control do you prefer to have? (e.g., detailed settings for each server/channel, simple on/off switches)
13. How important is it for you to have different notification settings for different times of the day or activities (e.g., work hours vs. leisure time)?

14. Is there anything else regarding Discord notifications you would like to mention that we haven't covered?

15. Any further thoughts on how to refine Discord's notification system?

17.2 Appendix: Needfinding Survey Questions

Participant Information

1. How old are you? *[Under 18, 18-24, 25-34, 35-44, 45-54, 55-64, 65 or older]*

2. What best describes your primary use of Discord? Please check all that apply. *[Gaming, Education, Work/Professional, Socializing, Other]*

3. If 'Other' was selected in the previous question, please specify: [*Open-ended*]

4. How often do you use Discord? *[Daily, Several times a week, Weekly, Less frequently]*

Notification Experience

5. On average, how many Discord notifications do you receive per day? *[1-10, 11-20, 21-50, 51-100, More than 100]*

6. On a scale of 1 to 5, rate your satisfaction level with the current Discord notification system. *[1 = Very Unsatisfied, 2 = Unsatisfied, 3 = Neutral, 4 = Satisfied, and 5 = Very Satisfied]*

7. Have you ever missed important messages due to the volume of notifications? *[Yes, No]*

8. If answered 'Yes' to the previous question, please briefly describe the situation. *[Open-ended]*

Notification Preferences

9. What types of Discord notifications do you find most valuable? (Select all that apply): *[Direct Messages, Server @mentions, Role @mentions, New posts in watched channels, Friend Requests, Other]*

10. If 'Other' was selected in the previous question, please specify. [*Open-ended*]

11. What features would you like to see in a redesigned notification system? (Select all that apply): *[More granular control over notification types, Ability to prioritize notifications, Options to mute notifications based on keywords or topics, Better organization of notifications (e.g., categorization, summaries), Notification filters (e.g., filter by server, role, or keyword), Smart notifications based on time of day or activity (e.g. "Do Not Disturb", "Sleep"), Other]*

12. If 'Other' was selected in the previous question, please specify. [*Open-ended*]

13. On a scale of 1 to 5, rate the importance of having customizable notification settings tailored to different contexts (e.g., work, gaming, studying). *[1 =Not important, 2 = Somewhat Important, 3 = Neutral, 4 = Important, 5 = Extremely Important]*

Additional Feedback

14. Please share any specific challenges you have faced with Discord's notification system and any ideas you have for addressing them. *[Open-ended]*

15. Do you have any additional comments or suggestions for the Discord notification system redesign? *[Open-ended]*

17.3 Appendix: Needfinding User Interview Notes

Raw user interview responses are stored in a text file and that can be found in the following links:

17.3.1 *Interview 1*

https://gtvault-my.sharepoint.com/:t:/g/personal/bwatts38_gatech_edu/ERboNKodzq9Mk6hWrI-E8v4BCfC6cyCKEKTSNKH7IGKMcA?e=zQ1gzh

17.3.2 *Interview 2*

https://gtvault-my.sharepoint.com/:t:/g/personal/bwatts38_gatech_edu/ES5ggwOHz2xEvBtxTYaPIxQBV3SHctd9MFr-cuCR6Dmdew?e=Nlm6Cc

17.3.3 *Interview 3*

https://1drv.ms/t/s!ArAwipZbPDVVg4YNviV_FIanAsbqAQ?e=vRKH4d

17.3.4 *Interview 4*

https://docs.google.com/document/d/1ttBfGogyu_dJIcqntc2LWw30Al_S1_u_oUVt5KGMClo/edit?usp=sharing

17.3.5 *Interview 5*

https://docs.google.com/document/d/1TK4-eLoXCM7rIKm42Ju_Pd6D4wV05WuoyApePouGEBQ/edit?usp=sharing

17.4 Appendix: Needfinding Survey Responses

Raw survey responses are stored in a csv file that can be found in the following link: https://gtvault-my.sharepoint.com/:x:/g/personal/bwatts38_gatech_edu/EW500MbNVzpAiENOqi_4DL8BA3_-nKjEisSFErv42N6R4Q?e=Oh7cA0

17.5 Appendix: Needfinding Heuristic Analysis

Below the in-depth analysis focused on server and channel notification settings along with the notification signals categorized by heuristic principles:

1. **Visibility of System Status:**

   Good: Discord's use of visual indicators like the red dot and number on server icons and channel names effectively communicates new notifications. This ensures users are constantly aware of unread messages, aligning well with the principle of keeping users informed about system status.

   Bad: The effects of muting options, particularly the server-wide mute, are not immediately apparent. Users might not fully understand that enabling this option will suppress all notifications from the server, except for direct mentions. This lack of clarity can lead to misunderstandings about the system's current state.

2. **User Control and Freedom**

Good: The system empowers users with the ability to mute entire servers or specific channels, providing a significant degree of control over the influx of notifications. This feature is particularly useful for users who participate in multiple servers and wish to minimize distractions.

Bad: The system's granularity in muting settings is limited. Users cannot mute specific types of notifications within a channel (e.g., only @mentions or message reactions), which restricts their ability to tailor the notification experience precisely to their preferences.

3. **Flexibility and Efficiency of Use**

   Good: Discord allows for notification customization at the channel level, enabling users to prioritize notifications from channels that are more important to them. This level of customization is beneficial for users who need to manage their attention across various conversations.

   Bad: The current setup lacks efficiency-enhancing features like notification setting presets or shortcuts that could allow users to quickly apply a set of preferred notification settings across different channels or servers. This shortfall impacts power users who interact with many channels and servers daily.

4. **Error Prevention and Recovery**

   Good: The mute duration feature is a thoughtful addition that helps prevent users from accidentally muting a channel or server indefinitely. By allowing users to set a time limit for the mute, the system provides a safety net for recovering from potential errors.

   Bad: There's a noticeable absence of interactive feedback or guidance when users configure their notification settings. This can lead to errors in setting preferences, especially for users who might not be familiar with the implications of their choices. The system could benefit from more explicit confirmations or warnings about the effects of certain settings.

5. **Recognition Rather Than Recall**

   Good: The notification system's use of universally recognized icons (e.g., the bell for mute/unmute functions) and consistent visual cues (e.g., red dots for

new messages) aids users in identifying functionalities without having to remember them.

Bad: Users are expected to remember their customized notification settings across various servers and channels. The lack of a consolidated view or summary of all notification preferences forces users to recall their settings, increasing cognitive load and the potential for confusion.

6. **Aesthetic and Minimalist Design**

   Good: The design of the notification settings interface is clean and straightforward, with options presented in a manner that avoids overwhelming users with excessive information.

   Bad: Despite the generally minimalist design, the 'Notification Overrides' section can become cluttered, especially for users who are part of many channels and servers. The long list of customizable options, while powerful, can detract from the simplicity and clarity of the interface.

User Scenarios and Personas in Context

Considering specific user scenarios and personas adds depth to our analysis:

- **Emma's Scenario**: As a college student, Emma values Discord for both academic collaborations and socializing. However, she often misses crucial updates from her study groups amidst the flood of social notifications. This scenario highlights the need for improved notification prioritization and filtering, allowing users like Emma to focus on essential communications without completely disconnecting from her social circles.
- **David's Scenario**: Working remotely, David relies on Discord for team communications but finds the mute settings challenging. He's missed important updates after forgetting to unmute channels, reflecting a gap in the system's ability to prevent and recover from user errors. Enhancing visibility and feedback regarding mute statuses could mitigate such issues, making the system more forgiving and user-friendly.

17.6 Appendix: Brainstorming Ideas

Ideas are all placed on a whiteboard linked here:

https://whiteboard.office.com/me/whiteboards/p/c3BvOmhodHBzOi8vZ3R2YXVsdC1teS5zaGFyZXBvaW50LmNvbS9wZXJzb25hbC9id2FodHMzOF9nYXRlY2hfZWR1/b!YDguXp4ytUuzIdRbO1ehxyLpe2ACu-VCtjIxsLZYKFE-5XGAU6xJTZ9p_FWNOU0Q/01CTMH2GLENXWLUGG7A5BL3LTGXCQEZONM

17.7 Appendix: Initial Prototypes Survey Questions

PROTOTYPE 1: Smart Modes

1. Smart Modes allow users to create custom notification settings like "Work," "Study," "Gaming," etc. Go to the link to view the provided material for this prototype before answering the following questions. [*insert link here*]. *[I have accessed the provided link; I have not accessed the provided link]*
2. What is your initial reaction to the Smart Modes feature? [Open-ended]
3. Is the concept of having customizable modes for different activities something you find valuable in managing notifications? *[Yes/No]*
4. Rate the statement: "Identifying and switching between different Smart Modes was straightforward." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*
5. Rate the statement: "Setting up and customizing my Smart Modes felt intuitive and hassle-free." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*
6. Please share any thoughts or suggestions you have regarding Smart Modes. *[Open-ended]*

PROTOTYPE 2: Content Filters

7. Content Filters give users the ability to set notification preferences based on specific keywords, phrases, or categories. Go to the link to view the provided material for this prototype before answering the following questions. [*insert link here*]. *[I have accessed the provided link; I have not accessed the provided link]*
8. Share your initial thoughts on the Content Filters feature. *[Open-ended]*
9. Do you think being able to filter notifications based on keywords or content is beneficial? *[Yes/No]*

10. Rate the statement: "The feature to create and apply Content Filters was clearly presented in the settings." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

11. Rate the statement: "Configuring my filters and understanding their impact on my notifications was intuitive." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

12. Please share any concerns or ideas you might have about the Content Filters feature. *[Open-ended]*

PROTOTYPE 3: PriorityHub

13. PriorityHub lets users mark certain contacts or channels as high priority, affecting the visual and audio cues for their notifications. Go to the link to view the provided material for this prototype before answering the following questions. [*insert link here*]. *[I have accessed the provided link; I have not accessed the provided link]*

14. Describe your first impression of PriorityHub. *[Open-ended]*

15. Do you find the idea of setting priority levels for contacts and channels useful? [Yes/No]

16. Rate the statement: "The visual cues for high-priority notifications were clear and helped in recognizing important alerts." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

17. Rate the statement: "Managing priority contacts and channels within PriorityHub was a straightforward process." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

18. Please provide any feedback or recommendations you have for PriorityHub. *[Open-ended]*

GENERAL QUESTIONS: Applicable to All Prototypes

19. Rate the statement: "The proposed features would significantly improve the way I handle notifications in Discord." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

20. Which prototype do you feel would best integrate with your current use of Discord? *[Options: Smart Modes / Content Filters / PriorityHub]*

21. What do you like most about the prototype you chose in the previous question, and why? *[Open-ended]*

22. Are there any additional comments or insights you would like to share about any of the prototypes? *[Open-ended]*

17.8 Appendix: Initial Prototypes Survey Responses

Raw survey responses are stored in a csv file that can be found in the following link:
https://docs.google.com/spreadsheets/d/1RIKIbeIopxHfAQ5qIytQa9QaU7XAoyJiVGM4EXYUAw4/edit?usp=sharing

17.9 Appendix: Final Prototype Survey Questions

1. *Q1*

2. Rate the statement: "The proposed features would significantly improve the way users manage notifications in Discord." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

3. Rate the statement: "The proposed features would significantly offer users more direct control over the notifications deemed important to them." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

4. Rate the statement: "The proposed features would significantly provide a more nuanced and user-friendly approach to proactively prioritize notifications from channels and contacts users genuinely care about." *[Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree]*

5. What is your opinion on the ability to prioritize a server?

6. What is your opinion on the ability to prioritize a channel on a server?

7. What is your opinion on the ability to prioritize a DM?

8. What is your opinion on the ability to look at your prioritized messages inbox?

9. What is your opinion on the ability to view prioritization settings?

10. Do you believe the default setting for all should be low prioritization? [Yes/No]

11. What are some criticisms of the proposed features?
12. What is your favorite feature of the final prototype?
13. What is your least favorite feature of the final prototype?
14. What are some improvements to our design we could incorporate?
15. Overall, what would you assess our final prototype? *[Scale: 1 = Terrible, 2 = Needs improvement, 3 = Neutral, 4 = Good, and 5 = Great]*
16. Why?

17.10 Appendix: Final Prototype Survey Response

Raw survey responses are stored in a csv file that can be found in the following link: https://1drv.ms/u/s!ArAwipZbPDVVlJwTkdoSXFJl1PdeaA?e=lGGwD3