[Insert index here!]

Analysis

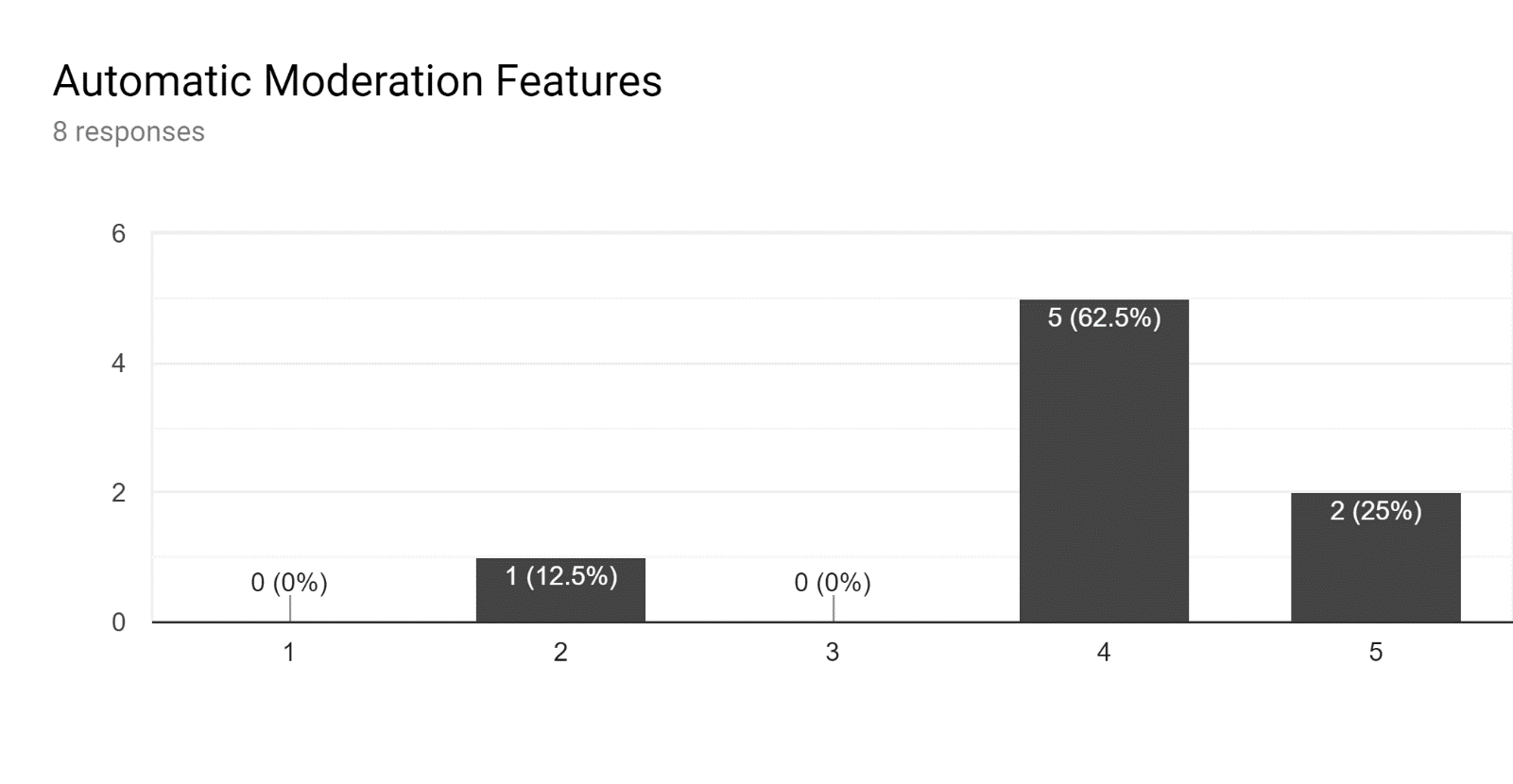
Project Background

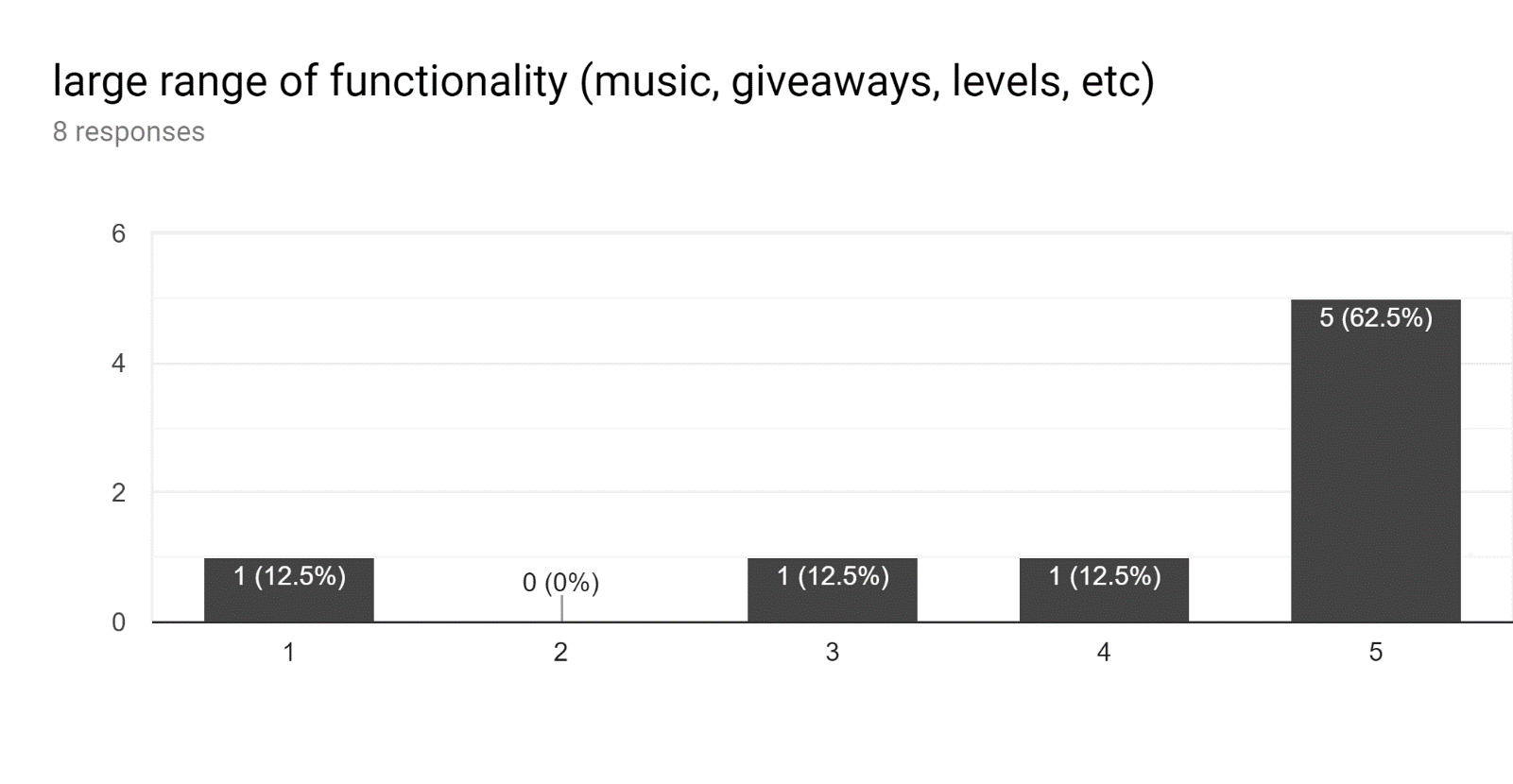
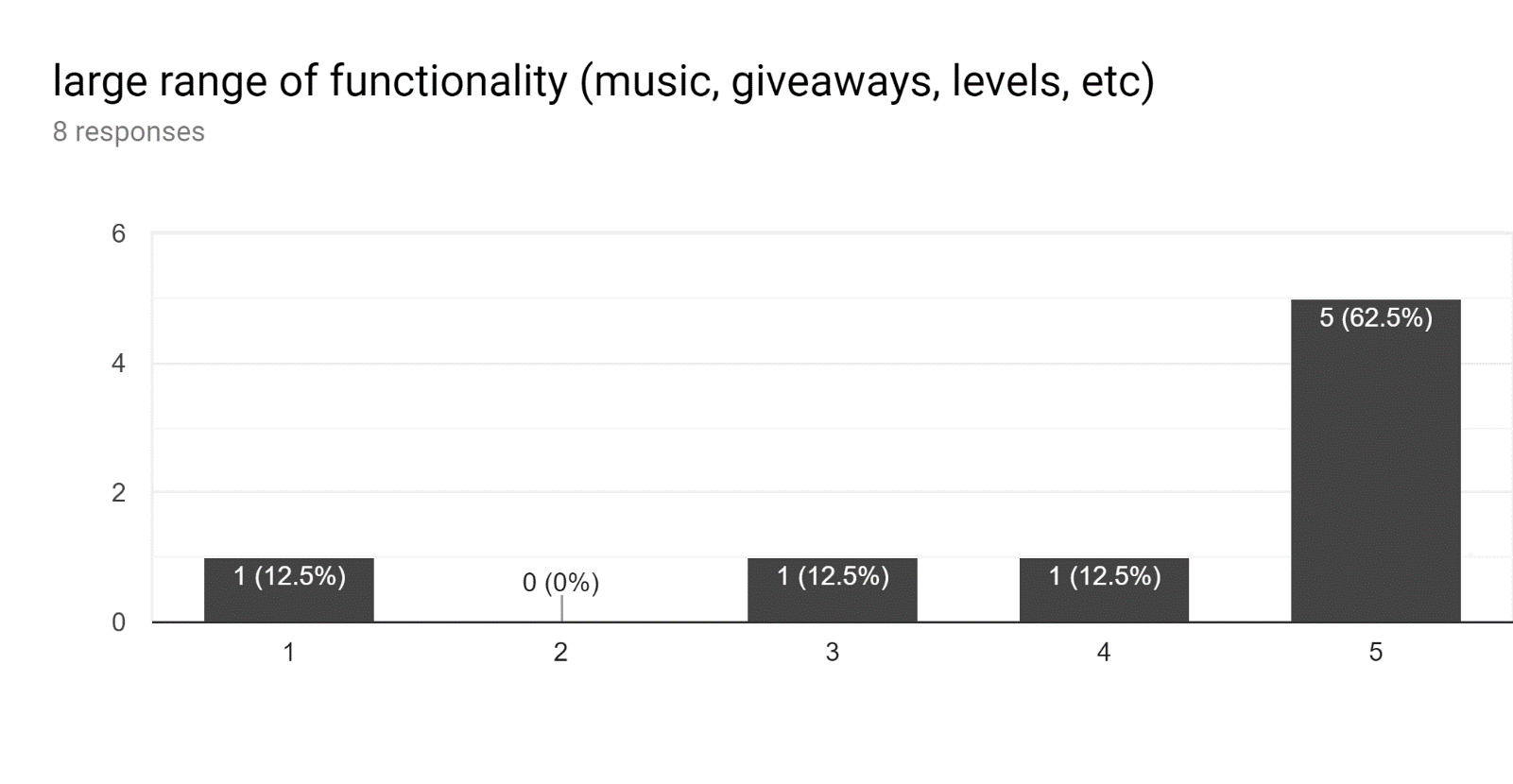
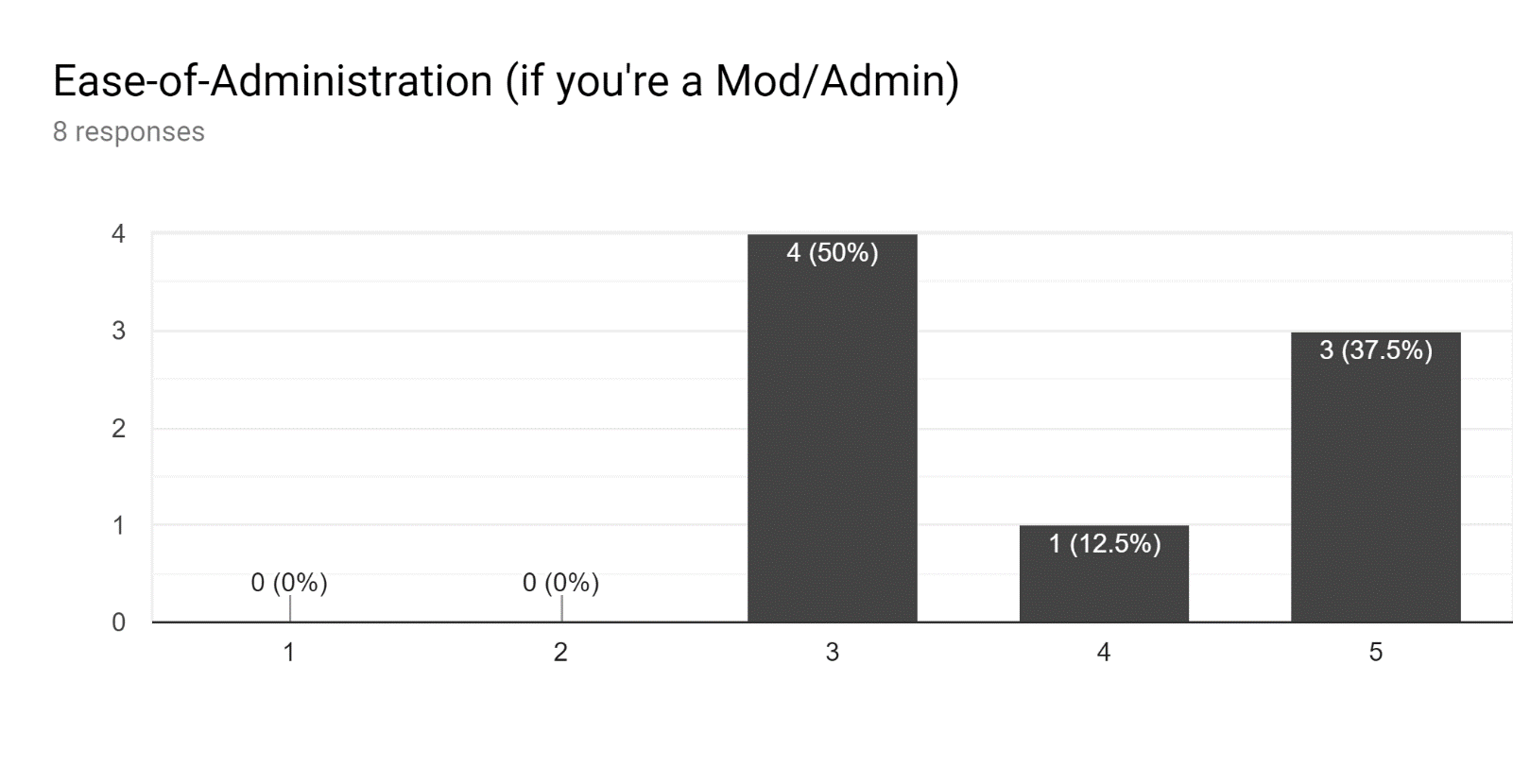
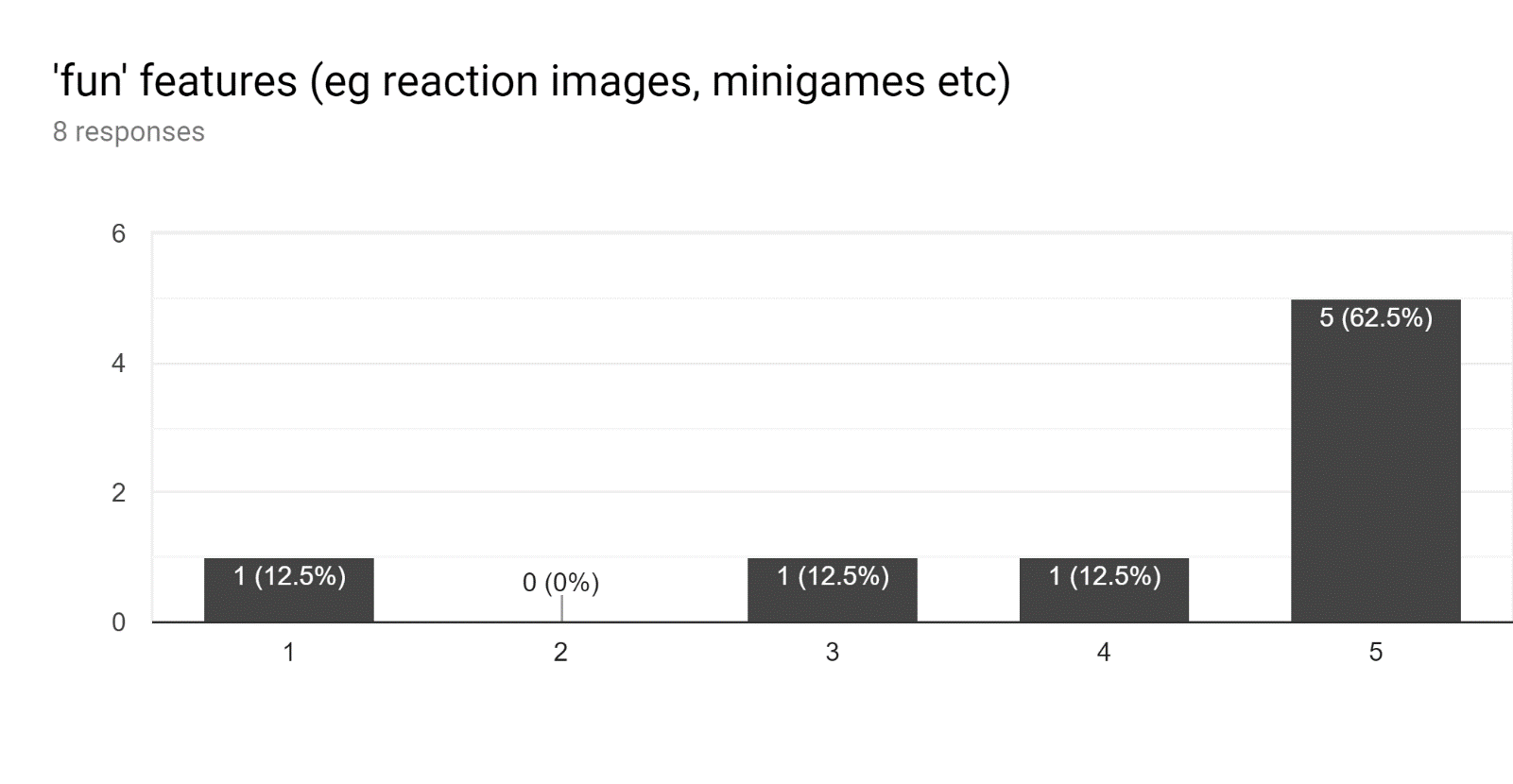
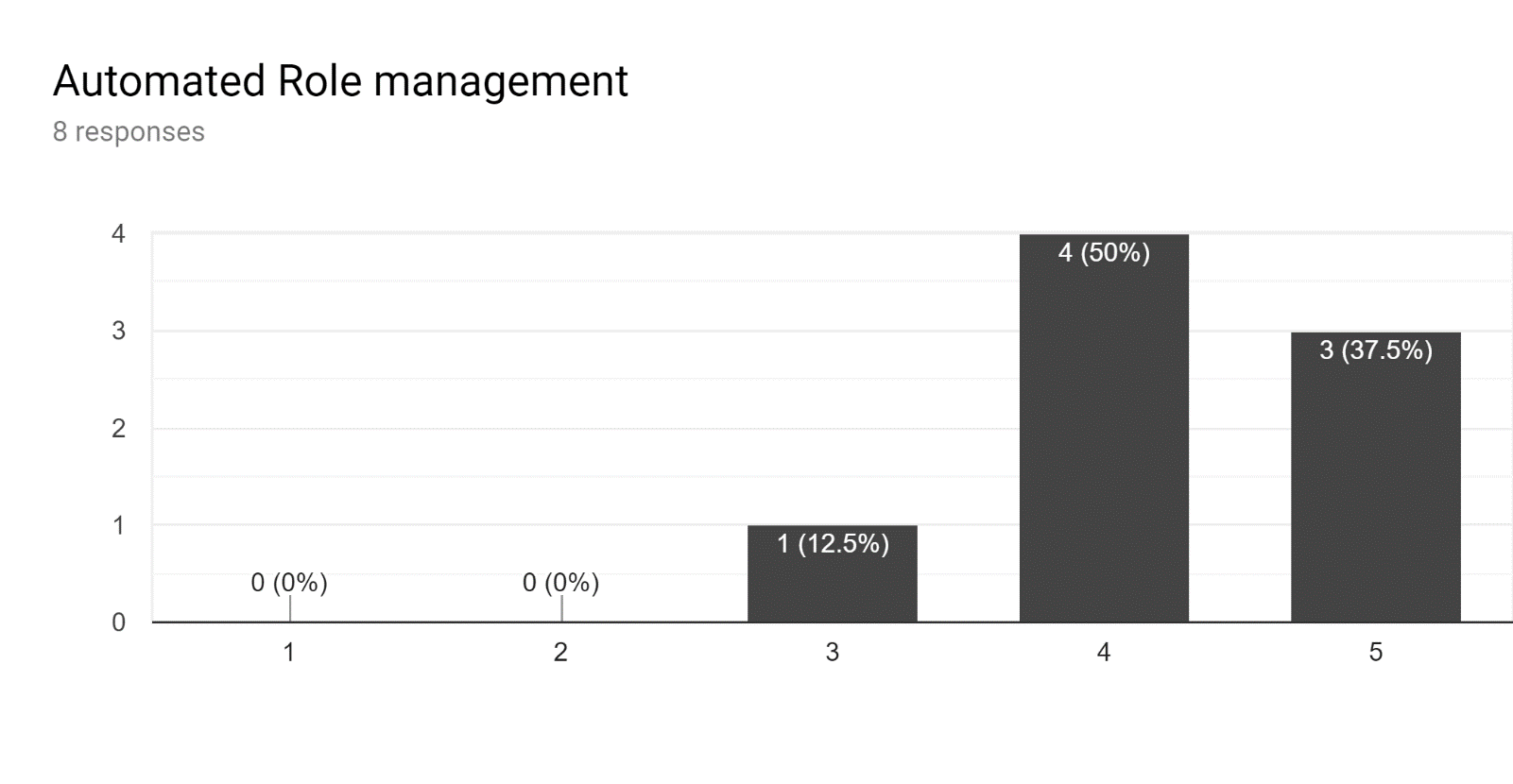
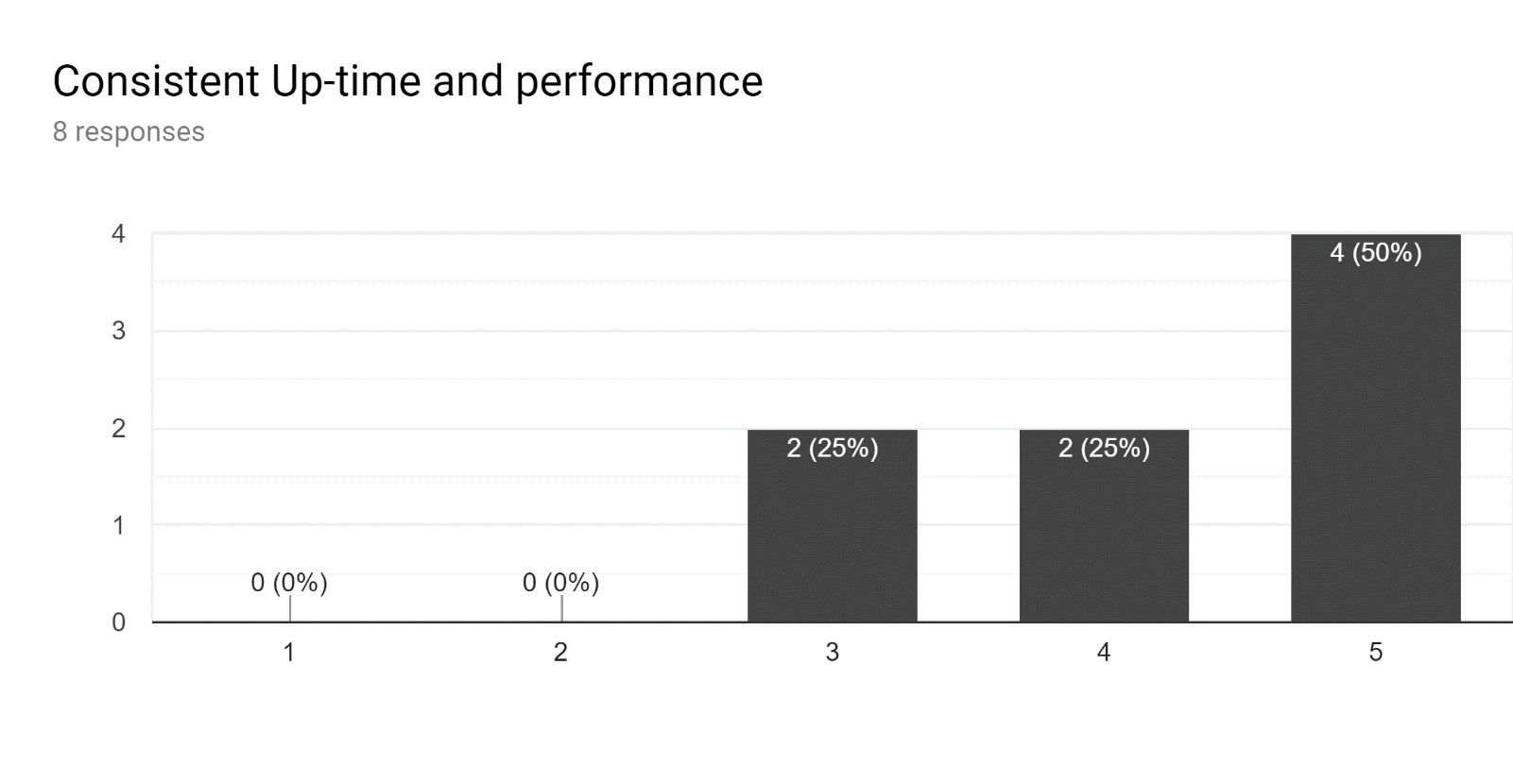
Over the last few years, many people like myself have started using the now very popular Social Media Platform Discord. For those unaware, Discord is a massively popular communication app, that allows for ‘servers’ that contain both text and voice channels. Users can join these servers and are managed by a rather complex permissions system as to fine tune what users can do what where, oftentimes governed by ‘tags’ called roles. Due to the large numbers of users, manual moderation quickly became inefficient. In response, discord released a REST/WebSocket based API which people used to build projects such as artificial users (‘bots’) coded in various languages. Overlord is one such Bot and seeks to implement features that often require multiple bots into one convenient and powerful package with the aims of being able to be easily selfhosted and modified by end users seeking to add custom functionality. To this end, I will be aiming to package it into a docker container. (see <https://www.docker.com/> for more information.)

My end users are both server administrators/moderators and generic server Users. To this end, I want to ensure that the management backend is simple enough for first-time set and forget Admins, but also powerful enough for those seeking to add custom functionality, whilst being robust and easy to use for the other server members.

End user Survey

I created and distributed a simple survey on google forms, allowing people to answer on a scale of 1-5 (1: least important, 5: most) about what they think are the most important features of such a bot, as well as some open-ended questions for other functionality. I got a limited number of responses, but the trends were very interesting to examine in regards to my development priorities. It showed that potential end users place a higher demand on more simple Features (such as automatic moderation) than more ‘advanced’ features (such as the ability to add custom commands) which is behaviour that I’d expect given the 1:100+ admin : generic user split and the differing priorities of each group.





Existing Solution analysis

due to the popularity and the demographic of people that discord is aimed towards, there are quite a few existing solutions – some of which are far more popular than others. To prevent succumbing to “second system syndrome” as well as to evaluate what other solutions have done right, Using a popular discord bot ‘ranking’ website (<https://discordbots.org/list/top> - note this order isn’t on *actual* user count, rather on votes), here are overlord’s Top 5 Competitors:

1. MEE6 – this bot primarily focuses on ease-of-use and administration, focusing on polished ‘basic’ features over more advanced functionality. Possessing an easy Oauth dashboard for cross-guild management, as well as a very intuitive user interface. The overwhelming popularity of this bot further proves my earlier assertions that focusing on more ‘basic’ features will be a better use of my time.
2. Mudae – Mudae is primarily a ‘fun’ focused bot with no moderation features whatsoever, but the User Experience within discord using the bot is top-notch and is something I will definitely aim to rival with overlord. It again shows that User Experience (UX) is something I’m going to have to nail with Overlord.
3. Pokécord – Pokécord is primarily Minigame/’fun’ focused, much like Mundae (and some others in this list)

|  |  |
| --- | --- |
| Objective/feature | Criteria |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |