## User testing

As a rough estimate I wanted over 80% completion, for confirmation that there are next to no bugs, or issues with the main program. A score of 3 meant the task was completed with no issues, a score of 2 meant the task was completed with issues, and a score of 1 meant the user couldn't perform the task. The user testing scores are summarised as follows:

| | Task | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 2.7 |
| User 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| User 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 |
| User 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| User 4 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| User 5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 1 |
| User 6 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Sum | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 15 |
| % | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 83 |

Overall, as each task was completed with over 80%, it is safe to assume that my app runs to a near flawless state, which was the goal.

## User survey (quantitative)

As a rough estimate I wanted over 80% agreement, for confirmation on a good user experience. A score of 3 means the user agrees with the statement, a score of 2 means the user is neutral, and a score of 1 means the user disagrees with the statement. A summary of the quantitative data collected from the survey:

| | Statement | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| | Good user exp | Ease | Bug-less | Personalized | Originality | Unique | Applicability |
| User 1 | 3 | 2.5 | 3 | 3 | 3 | 3 | 3 |
| User 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| User 3 | 3 | 2 | 3 | 3 | 3 | 3 | 3 |
| User 4 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| User 5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| User 6 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Sum | 18 | 16.5 | 18 | 18 | 18 | 18 | 18 |
| % | 100 | 92 | 100 | 100 | 100 | 100 | 100 |

Overall, it's safe to assume that the users enjoyed their experience, and were happy with the app, as each statement has well over 80% satisfaction.

## User survey (qualitative)

The one question I will analyze is the question on how the users felt the app could be improved. 50% of the users wanted a way to select individual songs, just like they could select visualisations. Though this wasn't the only suggestion, it was the most requested. All users came from different backgrounds, and yet despite all their differences in experience, half of them agreed that having a selection menu for the tracks would be a useful extension.


## Reflection on user feedback

As a whole it's safe to say my app, in its current state, is reaching users satisfaction levels, as well as functioning with little to no problems. Some users didn't perfectly score task 2.7, which was the change of fill colour. It makes sense on my end, as that fill option only appears when in fullscreen, due to the scaling of UI elements. In hindsight it would've made sense to scale the appearance functionality as well, so there wouldn't be any confusion. Isolating those specific scores and comments shows that users didn't have a problem with the feature itself, more so a problem with when the feature was available. It's also worth noting some users didn't score "ease" perfectly on the survey, this ties in with the lower scores on task 2.7, and would be something to work on and improve in future iterations of this app.

It follows that the one feature most requested is a feature that improves the ease and usability of the app. Before doing the test, my biggest concern was lack of uniqueness and originality, and it was where I was spending most of my coding time improving upon. This feedback is eye opening, as what I overlooked seemed to be the issue a few users had. In an ideal setting I would like to conduct multiple tests and surveys during the development of the app, to find out what should be prioritized ahead of time, to minimize wasted time, and improve the general experience a user has when using my application.