Sprint 1 Retrospective

2 weeks, 11/12/2017 -> 23/12/2017

# Stories:

|  |  |  |
| --- | --- | --- |
| **Story** | **Final Status** | **Future Extensions** |
| User can sign up | On web page. Functional but minimal frontend GUI | Email validation, dedicated page or modal, and toast confirmation |
| User can log in | “” | Allow user to log in with email, have toast confirmation. For apps, have checkbox to store login information. Encrypt stored PC passwords |
| User can associate device with account | Done automatically when user logs in on a device | Allow user to revoke devices, think about logging in different users on the same device |
| User can scan music | Works for small datasets, but duplicate track bugs with large datasets. Pc app can select music folders to scan. | Fix duplicate track bug, allow multiple filepaths per track per device. Scan playlists. |
| User can see music | A sortable list is shown with all user music. | Show playlists, show unique tracks with summary of devices + paths, search bar |
| User can play music via youtube | Clicking on a track will trigger that YouTube to play music. | Have a play button or work on ui for individual tracks to make the functionality more obvious. |

|  |  |  |  |
| --- | --- | --- | --- |
| **task** | **date completed** | **exp. Hrs** | **act. Hrs** |
| node js server | 12-Dec | 3 | 0 |
| setup database for accounts | 12-Dec | 1 | 3 |
| authenticated account creation | 12-Dec | 8 | 13 |
| authenticated login | 12-Dec | 3 | 0 |
| react website | 13-Dec | 3 | 7 |
| pc app tray icon | 13-Dec | 5 | 2 |
|  | 14-Dec |  |  |
|  | 15-Dec |  |  |
| Day 3 | 16-Dec |  |  |
| Day 4 | 17-Dec |  |  |
| pc app login | 18-Dec | 8 | 2 |
| android app | 19-Dec | 5 | 2 |
| android login | 19-Dec | 5 | 1.5 |
| setup database for scanned tracks | 19-Dec | 2 | 2 |
| Day 7 | 20-Dec |  |  |
| server add scanned tracks | 21-Dec | 2 | 6 |
| pc app select music folders gui | 22-Dec | 3 | 2 |
| android app scan music | 22-Dec | 5 | 3 |
| pc app scan tracks | 23-Dec | 5 | 4 |
| website list music | 23-Dec | 5 | 3 |
| setup youtube api | 23-Dec | 3 | 1 |
| website play track youtube | 23-Dec | 5 | 4 |
|  | total | 71 | 55.5 |

## What went well:

* Finished all minimum viable product functionality.
* Functional locally.
* Have a good starting point with basic authentication for future products.

## What was painful:

* Finding ways to do authentication and having it work together on each platform.
* Combining business logic with database low level queries. E.g. inserting new tracks required 3 linked fk inserts but in some cases part of the data needed to be updated instead of inserted. No tidy way to do this with Sequelize
* Debugging in node js

## Next time:

* Record start and end times, not just hours worked on a task.
* A way to separate business logic and DB functionality.
* Research async / await to replace promises.
* Stories were a bit too general this sprint. Most of them can be worked on further. Either we keep them to use again next time and prefix them V2, or make more in detail stories next time.
* Add new tasks to Trello as we go (maybe in a new column?).
* Deploy the app.
* Think about getting a debugging environment / linting for node js / react
* Think about setting up some automated tests for the rest api