School of Computing
Final Year Engineering Project

Project Initiation Document

Johnny Morrison-Howe

UP926743

Podcast Scrobbler

1. Basic details

Student name:	Johnny Morrison-Howe
Draft project title:	Podcast Scrobbler
Course and year:	BSc Computer Science, Third Year
Project supervisor:	Dalin Zhou
Client organisation:	*N/A*
Client contact name:	*N/A*

2. Degree suitability

Whilst my project is perhaps more suitable for a Software Engineering degree as I will be writing a mobile(or multiplatform) application, I believe there is sufficient processing involved that will use some algorithms used in Computer Science, for instance the parsing of text and the sorting/ordering of podcasts, episodes and tracks.

3. Outline of the project environment and problem to be solved

In my time of listening to music, I've done a lot of two things that are incompatible: listening to mixes in the form of podcasts, and tracking the music I listen to with Last.fm (a service for logging the tracks you listen to, sharing with friends, and providing recommendations). When tracking a podcast, only the title is tracked, as this is what appears in most media player APIs – even though really this kind of podcast is filled with 10-20 tracks, which is what I *actually* want to keep record of. Tracklists are often found in the description of a podcast however, in the order in which they're played and sometimes with timestamps.

I'd like to design and build an application both for managing and playing podcasts, and for parsing tracklists from a description, and uploading the metadata when that track is played (either using timestamps or interpolation) using Last.fm's scrobbling API. (Scrobbling being the name Last.fm uses for tracking music).

4. Project aim and objectives

The aim for the project is to come out with a working native Android application able to manage and play podcasts, as well as tracking metadata using Last.fm's API. I'd like it to utilise an aesthetically pleasing design (Material UI 3 ideally) as well, as that'd be good practice for the industry.

As well as this, I'd like to implement unit testing and CI/CD functions. I can verify how well the algorithms I write work by testing using podcasts I hadn't seen before. If these produced usable metadata, then I could confirm that my project had been a success.

5. Project constraints

The main constraint for this project is the time it'd take to learn and build this kind of application. Although I have experience with Java and Flutter, I've never developed an entire application with just Java (or Kotlin if I choose to learn that).

6. Facilities and resources

I will only require my own development machine, toolchain (Android Studio), and an internet connection to write and build this application. I will also use my own Android device for testing out of convenience, however this is not mandatory.

7. Log of risks

Description	Impact	Mitigation/Avoidance
Last.fm decides to close or restrict its API	High	Although I wouldn't have much influence to avoid this issue, I would be able to compile and display the list of tracks on-device to take Last.fm out of the equation. This would defeat much of the point of the application, however it would still allow me a chance to build a Material UI 3 podcast playing app.
Parsing a podcast's description doesn't prove possible with enough podcasts (it's too variable)	Medium	Although it may be possible to implement looser pattern matching, a completely alternate solution would be using an API like ShazamKit to recognize the song and get track data for every song.
My laptop breaks	High	I plan to store all source code on an SCM like GitHub, therefore I should be able to still develop my application using the University machines (even if less convenient).

8. Project deliverables

I plan to produce:

- Project report
- Compiled application
- Source code and tests
- Documentation
- CI/CD Scripts

9. Project approach

I'll manage my project primarily using Excel and OneNote, to produce tables for time I should promote to different parts of the project, and to note recommended reading when managing smaller tasks, respectively. As the teamwork is not applicable to this project, I'll use a Waterfall methodology for the project, making use of Gantt charts to plan my time. Although strict Waterfall methodology requires that each step isn't started until the last step is completed, it would be advantageous to be writing notes for the next step during the current step – i.e. design notes before development commences, and notes on development before I start writing the report.

I need to do research on existing solutions, no matter how roundabout, to find out what I can solve/improve. As well as this, different frameworks and languages for the application; Even though I'm fairly set on using Kotlin and Jetpack Compose, this could change given other opinions. I also need to do research into design patterns, i.e. those found on https://m3.material.io/ . Starting points for these will often be found in documentation, however research papers supporting their choices would also be essential.

10. Project plan

My primary stages would be:

- Carrying out a literature review
- Establishing my requirements
- Planning individual components of artifacts
    - Ideally using SDK docs and design specifications
- Creating artifact
    - Setting up CI/CD pipelines, and SDKs
    - Obtaining API access for Last.fm
    - Writing and testing the actual artifact
    - Writing notes of my development experience for the report
- Testing my artifact's effectiveness at parsing podcast tracklists
    - I'll be using some podcasts to build and test the application, however it only truly works if it can parse something I haven't seen before or accounted for.
- Writing the report

As each task requires the one before it to be more or less complete, I'll be performing these tasks in order (although collecting notes during development to compile in the report). Whilst creating the artifact, I'll be self-teaching any programming languages/frameworks not known yet.

11. Supervisor Meetings

I've arranged fortnightly meetings with Dalin from 17:30-18:00 on Tuesdays (starting with the 25th of October). These will ideally be conducted face-to-face in his office, however it's been agreed that meeting remotely is also acceptable. We've also discussed extra sessions when necessary.

12. Legal, ethical, professional, social issues (mandatory)

Although I don't believe there are any professional or social issues, I will need to take security into account. Recording song metadata using Last.fm's API requires an account, therefore I will need to store the username and a private key for accessing the API. I plan only to store this information on the device however, so I would not need to deal with any data protection laws. Any end user other than myself would have a separate agreement with Last.fm for their own account on creation and when logging in, however I do not plan to conduct user research for this project.

Appendix A: Ethics certificate

The PDF is appended to the end of the document.

Appendix B: Gantt chart