# Disclaimer

This work is original and has not been previously submitted in support of any other course or qualification.

<*Type your name in full and sign below it, giving the date*.>

# Abstract

<*This follows the title page, on a single sheet of paper (can be single-spaced) and is a statement of the aims, method and results of your work i.e. is a summary of everything*.>

# Acknowledgements

<*It is courteous to express thanks for assistance given by your supervisor, external people, schools, etc*.>

# Contents Page

<*List Chapters with titles, and sub-sections if not too numerous.  Page numbers MUST be given.  Include a list of Appendices. The Table of Contents feature in Word makes this easy*>.

# List of Figures/Photographs

<*Photographs/tables/illustrations normally go in the body of your text.  If you want to use a figure or table, it must be labelled with a number and title.  Figure legends go below figures in the main text, and table legends go above tables. Remember to refer to the figure or table in the text*.>

# Chapter 1 Background and Motivation

## 1.1 Topic and background review

This project investigates applications for mobile devices which display chords and lyrics for musicians. These types of apps allow users to view the lyrics and chords of their chosen songs in the way that they are able to know how to perform them on a specific music instrument. Chords would be positioned on top of the specific words where they come into play in the song, that way the musician can know when to change to the next chord.

Some of the most common features are:

* Automatic scrolling, which allows the musician to view the whole song without having to let go of their instrument to scroll along.
* Clickable chords diagrams: chords are clickable, so they show a diagram on how to perform the chord when they are clicked.
* Ability to upload your own songs: Musicians can upload their own songs to the songs library of the app.
* Songs play along feature: The music of the song will be played for the user so the user can use it as a guide or even accompaniment in real time.
* Metronome: Some apps offer a built in metronome to allow the user control or visualize the tempo of the song.
* Courses and tutorials: some apps have available courses and tutorials that the users can buy to learn or improve their instrument skills.
* Communities: Some apps provide access to their own communities of musicians where they are encouraged to share and view what other musicians upload.

This project consists of the implementation of an Android app of this type that features some of these most common characteristics, but the most important feature is the ability to allow the user to upload a file downloaded from the iOS app “OnSong”, as well as from other platforms, and to import the sheet with the lyrics and chords to be read on our Android app. This is the feature that provides the highest value to our app and what we will build our business case around.

Another characteristic of our app is that it is open source. That means that the code is visible and accessible to other developers, and these will be able to modify it to make it better or adding extra features to the application. This will be directly under the supervision of the researcher and approval as owner of the project.

## Personal Involvement

I have been myself a user of this type of apps for many years while learning to play guitar, and although probably my musical skills don’t match with how long I have been playing for, I have a good understanding of the features, user pain points, most important tasks and red routes on chords sheet apps.

That experience provides knowledge on how the app should look and function from the user perspective, which will definitely help me at the design and testing stages, as well as motivation to craft a product that I will be able to use myself in the future and meet the general standards of the industry.

## 1.3 Project management methodology

The project management methodology that I will be using for this project is Extreme Programming (XP).

Extreme Programming is a project methodology that falls under the Agile methodology. (Agile Alliance, 2018)

In summary, this methodology calls for short sprints where the development team perform a sufficient design based on user stories, then from a simple and flexible design the team starts developing right away starting with the high value features that have been appointed. Automatic testing is carried out to ensure that the code has no bugs or errors.

The main characteristics of Extreme programming are:

* You will focus first on the most important functional requirements for the customer.
* XP can work with teams of any size; in our case we are developing individually.
* XP is flexible and adapt quickly to changes in the requirements for the project.
* XP calls for short release cycles and uses one-week sprints or iterations based on user stories to create most important requirements.
* Highest priority features will be implemented first and the rest will be considered lower value, and they will be only produced after the release date if there is no time to produce them before.
* Features must be tested to catch defects and bugs early.

Beck, K., & Andres, C. (2004)

There are other methodologies or frameworks that fall under the Agile style and are very popular, like Kanban or Scrum.

We have chosen XP because it focuses on software development, the XP values and practices are built around creating an environment conducive to development, and development programmer habits that help them design and build code that’s simple and easy to change. (Stellman & Greene, 2015)

Other Agile methodologies as Kanban and Scrum have a different focus. Kanban focuses on managing workflow and eliminating the waste from their process. Scrum focuses on project management primarily: The scope of the work that will be done, when that work will be delivered, and whether the outcome of that works meets the needs of the users and stakeholders. (Stellman & Greene, 2015)

Given that we are a solo developer and that we are time constrained XP methodology was a suitable choice as we will benefit from the short weekly sprints and ability to modify the design and make changes in the development as we progress on the project due to incremental design (see section 3.0).

Because we are not experienced developers, focusing on just some functionality each sprint and dividing them into higher and lower value will help us differentiate between more and less important tasks and help us achieve the core functionality of the program within our time constraints.

### 1.4 Initial functional Requirements

Although we are determining many of the product functional requirements based on user stories through the development phase in our sprints, there are some requirements (higher value features) that we already know that our product must contain:

FR1. Users will be able to upload a song to their personal library in the app and edit it.

FR2. Users will be able to import an iOS file and the app will display the chords sheet.

FR3. Users will be able to use automatic scroll when displaying a song.

FR4. Users will be able to classify their sheets and create custom sets.

### 1.5 Initial accessibility requirements

AR1. Users will be able to resize the chords sheet.

AR2. Users will be able to change icons and font size.

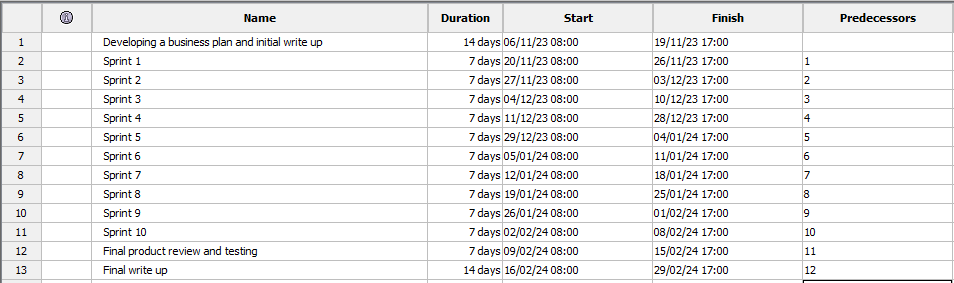
AR3. Users will be able to switch between light and dark mode.

### 1.6 Preliminary schedule and milestones

1. Developing a business plan and initial write up: 2 Weeks.
2. Developing phase based on weekly sprints: 10 weeks.
3. Final product review and testing: 1 week
4. Final write up: 2 weeks.

Total: 15 weeks

Total weeks until 20th of march (from 06/11/2023): 19 weeks



A screen shot of a diagram

Description automatically generated

# Business plan

#### 2.1 Business summary

The business consists of selling the users the possibility to remove the ads from our application.

The app will be available to download from the google play store for free. Users will be able to use the main feature of the app for free, but the app will contain adds. If the users want to remove the adds they will have to pay either a one-time purchase or a subscription.

#### 2.2 Business aims

The aims of the business are:

* Provide valuable features for the users.
* Provide an outstanding User Experience and Usability.
* Create customer loyalty.
* Have a good image and perception in the sector of apps for musicians.

### 2.3 Reasons

The reasons for doing this project are:

* Having identified a gap in the market of apps for musicians, which is the small number of apps that allow importing song sheets, in an easy, straight forward way, from different platforms into an android app.
* Expected benefits of developing a successful app that is able to gain a portion of the market.
* Low maintenance due to no backend required so only costs the project will generate will be during development which reduces the risks for carrying out the project considerably.

### 2.4 Market research

There are many Android apps for replacing paper chords sheets already in Google Play Store.

Many apps focus on teaching how to play guitar or piano, more than being a simple chords sheet. We will look at the ones that focus on acting as chords sheet databases as they compare better with our own app and the aim of our project.

The most popular ones (by number of downloads) are:

1. **Ultimate Guitar: Chords & Tabs (10M+ downloads)**

A black square with yellow letter g and horns

Description automatically generated

A screenshot of a computer

Description automatically generated

This is probably the most complete and polished app of this type for Android, being also available for iOS and Windows. It offers a free version. The free version shows adds and won’t allow users to use some of the premium features such as auto scrolling on android, being this the most useful feature and drastically forcing the user to buy the pro version. Also it doesn’t allow free users downloading the sheet to pdf and access to official tabs made by professional musicians.

It also provides a different type of subscription that allow the user access to courses to learn to play guitar and specific songs.

For the “Pro” Plan users are charged £24.99/month or £6.25/month if they pay for a whole year up front.

A screenshot of a music tab

Description automatically generated

(Google LLC, n.d.-c)

1. **Guitar Songs (10M+ downloads)**

A sheet of paper with writing on it

Description automatically generatedA screenshot of a music application

Description automatically generated

This app provides a more simplistic way to view the songs and also upload your own, it does not count with many features, but it provides the user with the most important ones such as auto scrolling, user personal library, transposing and metronome.

This app shows adds in abundance and there are two ways of removing these for the user: You can choose to remove the ads for a week by watching a long advertisement or you can make a one-time payment of £3.29 to disable the adds indefinitely.

1. **Guitar, ukelele & piano chords (Chordify) (5M+ downloads)**

A screenshot of a music tab

Description automatically generatedA screenshot of a music website

Description automatically generated

This app allows the user to play their instrument along with the song. The chords will be displayed and placed in a diagram that shows in real time when to play them on guitar, piano or ukelele. The difference between this app and the former two is that the lyrics are not shown but instead the app offers a clearer and dynamic visualization of the tempo and rhythm of the instrument that we wish to play.

(Google LLC, n.d.-a)

The approach this app applies for revenue generating is offering a free version with basic features and a premium version with more advanced features. The premium features are:

* Unlimited access to songs
* Transposing
* Download chord MIDI files
* Chord diagrams and sheets in PDF
* Change the tempo
* Setlists to organize songs
* No adds
* Loop parts of a song
* Digital capo
* Count-off for perfect timing
* Song and chords volume
* Upload your own music files

The price of the premium plan is £1/month when billed yearly or £6.59/month when billed monthly.

(Chordify , n.d.)

These were three of the most popular apps on the market but none of them allow users to upload their own iOS file of a song sheet, at least on the free version. Also, the previous apps offer a large database of songs available to the users. If we try to find apps that don’t offer a database of songs but instead just allow the users to upload their own songs and importing files, then there are not many, hence the gap in the market that we are trying to fill. But there are some that fulfils this purpose:

1. **SongbookPro (+100k downloads)**

A black and white square with a musical note in it

Description automatically generatedA screenshot of a website

Description automatically generated

A screenshot of a phone

Description automatically generatedThis is the most popular app with the characteristics that we were describing previously. It has been downloaded more than 100.000 times and it has a high rating within the Google Play Store. The basic features that this app offers are:

* Auto-scrolling and customizable auto-scrolling speed
* Metronome
* Sharing sheets
* Zooming in and out
* Drawing on the sheet with different colours
* Importing, uploading and editing a song
* Transposing
* Change of key

(Google LLC, n.d.-b)

The only two features that the premium plan unlocks are:

* Cloud synchronization for multiple devices
* Saving more than 12 songs in our library

The price of the premium plan is a one-time payment of £6.99. (SongbookPro Systems Limited, n.d.)

1. A black musical note on a white background

   Description automatically generated**OpenSongApp – Songbook (+100k downloads)**

A close-up of a logo

Description automatically generated

A screenshot of a phone

Description automatically generatedThis app is very similar to the previous one and implements these features:

* Auto-scrolling
* Looping control
* Transposing
* Zoom in and out
* Metronome
* Different app modes (Stage, performance and presentation)
* Creation of songs sets
* Importing, uploading and editing songs
* Wide app customization options

This app is completely free, and it does not contain advertisements and it does not aim to generate revenue which could be decisive factor for users when choosing this app over others for regular use.

However when we look at this app more closely and interact with the menus, we can see that is not as polished as the previous app, it feels messier, worse looking and menus are cluttered with information.

This app is currently not being updated and the last version was release in December 2022, they are not currently working on improving features or fixing bugs. The developers are now working in the version V6 beta that will be released soon and will add new features and user options apart from making the UI more intuitive and tidier.

(OpenSongApp, n.d.)

#### Market research conclusion

Looking at the market of apps for replacing chords sheets we can find many different types of them, but there are not many that satisfy the same need we are trying to fulfil: easily importing an iOS file and reading the chords sheet.

As well as that, we have identified some key features and attributes that have become standard in the industry of chords sheet replacement app for musicians:

* Auto scrolling.
* Clear and easy to use user interface.
* Transposing chords.
* Creation of custom sets or lists of tabs.
* Scaling the tab content to make it larger or smaller.

We are required to implement these features in order to be able to compete with the other apps in the sector.

Looking at the number of downloads for both of the apps that we will consider our main competitors (OpenSongApp and SongbookPro) we can confirm that there is a big enough market for us to be able to enter and gain a portion of it.

In order to achieve this we will aim to offer a similar or higher product quality at a lower price.

### 2.5 Business options

#### 2.5.1 Create a large and complex app

Having identified what are the main and most successful applications in the market, we could argue that they are the ones that count with an online database of song sheets, provide complex features apart from the basic ones and also provide some form of educational or interactive content to help the user learn how to play the instrument.

We could chose producing an app of this style, but the investment would be much higher than producing a more simplistic app with no backend and the benefits would not be expected to increase accordingly as we would be competing with the top companies that count with large teams.

Also our timescale is limited to 19 weeks so we would incur on the risk of trying to develop too complex and too many features and not reaching the quality standard that we need to achieve in order to attract and maintain customers.

#### 2.5.2 Create a simple app

Being our priority and highest value feature to provide an app that can import an iOS file and read it, we could just focus on creating an app that offers this feature and other basic features to the highest quality standard.

The investment and also the maintenance needed for this option would be much lower than in the previous option and we would be able to captivate users who are actually only interested in an electronic sheets notebook and don’t need an online catalogue of sheets. From our market research we know that at least we would have a market of +100,000 users, because that is the number of users that have downloaded the 2 most popular basic electronic song sheets that allow importing and reading files from other platforms.

#### 2.5.3 Not developing an app

Based on the market research we could also argue that the market offer is already high enough and there is no space for any more. In this case we decide to not take the risk of investing in our app and do not produce it.

### 2.6 Expected benefits

Creating a simple app is our chosen option, let’s look at the expected benefits of this option.

Our main competitor SongbookPro has more than 100,000 downloads with a review score of 4.7/5 so we can assume that the app has more than 100,000 users.

OpenSongApp, our second main competitor has more than 100,000 downloads with a review score of 4/5 so we can assume that the app has more than 100,000 users.

It is obvious that the market for our app is large, but also, we will be competing with other apps that have been in the industry for a long time, and they are high quality products.

Because we don’t know how many users we will get after we release our app, we will use different scenarios to visualize the possible outcomes of our product. The first scenario will consist of our Product having a poor performance on the market, in the second scenario it will have a medium performance and on the third scenario it will have an excellent performance.

To estimate how much money we could make from in-App advertising we will be working using the Google add services platform called Google AdMob. They provide us with a potential annual revenue based on the number of Monthly Active Users.

With an average of 50,000 monthly active users, we would obtain a potential annual revenue of $1,338 (Google LLC., 2019)

Based on this reference, we would get £0.021 per active monthly user.

### 2.7 Timescale

Project timescale (see 1.6):

The project is expected to be carried out over 15 weeks, with the possibility of extending the development another 4 weeks.

For Gantt Chart of the project refer to section 1.6

For the benefit time scale we will use a 3 year range.

### 2.8 Costs

We will take the average salary of a Graduate Software Developer in the UK as a reference for calculating the cost of the development hour.

The average salary for a Graduate Software Developer is £35.871 per year. That means that the average hourly salary for a Graduate Software Developer is £17.25 (standard 52 weeks worked per year, 40 hours a week) (Glassdoor LLC, 2023)

For marketing or advertising costs we will use Google Ads. With Google Ads, the most common way that the user gets charged for advertising is Pay-per-click (PPC). We will only get charged when an online user clicks on our add, if no users click on our adds then we won’t get charged. (Lyfe Marketing, 2021)

We can set up a budget so we can only get charged what our budget allows.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Scenario 1 - Poor performance – Low marketing investment** |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | **Year 1** | **Year 2** | **year 3** |
| **Costs** |  |  |  |  |  |
|  | **Development time (hours)** |  | 250 | 50 | 50 |
|  | **Development cost** |  | £ 4,312.50 | £ 862.50 | £ 862.50 |
|  | **Marketing** |  | £ 300.00 | £ 500.00 | £ 800.00 |
|  |  |  |  |  |  |
| **Total costs** |  |  | **£ 4,612.50** | **£ 1,362.50** | **£ 1,662.50** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Scenario 2 - Medium performance – Medium marketing investment** |  |  |  |  |
|  |  |  | **Year 1** | **Year 2** | **year 3** |
| **Costs** |  |  |  |  |  |
|  | **Development time (hours)** |  | 250 | 100 | 100 |
|  | **Development cost** |  | £ 4,312.50 | £ 1,725.00 | £ 1,725.00 |
|  | **Marketing** |  | £ 300.00 | £ 1,000.00 | £ 5,000.00 |
|  |  |  |  |  |  |
| **Total costs** |  |  | **£ 4,612.50** | **£ 2,725.00** | **£ 6,725.00** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Scenario 3 - Excellent performance – high marketing investment** |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | **Year 1** | **Year 2** | **year 3** |
| **Costs** |  |  |  |  |  |
|  | **Development time (hours)** |  | 250 | 100 | 100 |
|  | **Development cost** |  | £ 4,312.50 | £ 1,725.00 | £ 1,725.00 |
|  | **Marketing** |  | £ 300.00 | £ 5,000.00 | £ 15,000.00 |
|  |  |  |  |  |  |
| **Total costs** |  |  | **£ 4,612.50** | **£ 6,725.00** | **£ 16,725.00** |

* 1. Investment appraisal

In this section we will analyse the costs of the project against the expected benefits to evaluate possible results.

We have created 3 different scenarios where costs vary depending on marketing investment and maintenance and improvement development time in ongoing years.

The time scale for the costs and benefits is 3 years.

In terms development time, in all three projects we have the same number of development hours in year 1. This is because that is the estimate time that we will take into developing and publishing our product. Then, depending on the performance of the product in the market and the revenue generated we will invest more time the next 2 years on improving the product using part of the profits. The more profits we generate, the more money we can use to improve the product as it shows in the different scenarios.

The same concept is applied to marketing investment. The more money we generate, the more we can use to advertise our product and keep gaining users.

For the user to access the add free version of our product, and as our main income generation, we have chosen a low value subscription consisting in £0.99 a Month. This price that is set up with the intention of encourage users to subscribe being the amount so low, but also the subscription will allow us to keep receiving income from users over the time, until they unsubscribe.

We have estimated that an average of 20% of active users will subscribe to the add free version of the app.

* Scenario 1 – Poor performance, low marketing investment

A screenshot of a spreadsheet

Description automatically generated

* Scenario 2 – Medium performance, medium marketing investment

A table with numbers and text

Description automatically generated

* Scenario 3 – High performance, high marketing investment

A screenshot of a spreadsheet

Description automatically generated

As we can see from the estimated revenue in the different scenarios, in only the first scenario we would get losses, and over time they would not be significantly high, being able to adjust the development time that we use in maintenance or improving the product, as well as the investment in marketing.

### Major risks

|  |  |
| --- | --- |
| **Possible risk** | **Response/s** |
| Not meeting the project deadline | * Dividing the App functional requirements in high value and low value features, and only work on the low value features once all the high value features have been developed * Having left 4 weeks of backup time in case we need more development time, or in case we want to add low value features or minor tweaks or changes |
| The product quality not reaching the market standards due to lack of experience or skills | Adjusting the price of the product in accordance to ensure that we can gain a portion of the market |
| The demand of the project not being what was predicted | Promoting the product using online marketing on social media or relevant musician sites |
| Low productivity | Ensuring time management and spreading the work hours wisely to avoid burnout  Stay motivated |
| Bad response to the product from the users | * Testing the software thoroughly to avoid bugs and issues * Performing users surveys and user testing to find issues and malfunctions before the product is released |

# Design

## 3.0 Introduction

For Implementing our system we have chosen the Extreme Programming methodology (XP). See section 1.3 for a general overview on this methodology.

In XP, when designing systems, Incremental Design is implemented.

In software development, the assumption that the cost of large-scale design changes dramatically increases over time has led to decades of thinking that the most economical design strategy is to make big design decisions early and defer all small-scale decisions until later. That affirmation may no longer be valid. Our work using XP is to create conditions under which the cost of changing the software does not rise dramatically. System testing, short development cycles or sprints and the well formulated stories all contribute to keep the cost of the changes low.

XP teams need to keep the design investment in proportion to the needs of the system so far instead of piling story over story as quickly as possible with the least possible investment in design.

In other words: when you invest enough in design over the short term or sprints, the less likely you will have to change the design in the future, and if changes need to be made, they will be cheaper to make because they will be lower level changes. Keeping the design simple and avoiding duplication are best practices do make future changes in the design cheaper to make.

In conclusion, in XP, enough design investment done close to when it is used is a more efficient, flexible way to design systems that also make future changes in the design cheaper to make.

(Beck & Andres, 2004)

Following the XP implementation methodology we will be working on weekly sprints. We will base our sprints in user stories that will provide higher value functionality and lower value functionality.

In each sprint we will generate an user story that provides functionality that will takes approximately a week to be completed. We will design our app based on the user story in Figma and then we will build our app in Android Studio.

## Sprint 1

User story:

“As a chords app for musicians user, I want to be able to upload a song to my personal library.”

High value tasks:

* Task 1: Create a song class.
* Task 2: Create an upload form to save the details of each song.

Low value tasks:

* Create a recycler view to show the saved tabs in the main screen

A white rectangular frame with black border

Description automatically generatedA screenshot of a phone

Description automatically generated

Development of sprint:

A screenshot of a computer

Description automatically generatedThe high value tasks were completed within the week. However we didn’t have enough time to complete the lower value task (create a recycler view for the main screen). This task will be moved to the next sprint, being this the high value task and complementing it with other tasks, trying to aim to create a workload that we will be able to complete in another week sprint.

Some changes that occurred from the design to the actual development:

* Instead of switching between the general and advanced fields using two labels or buttons on top of the fields, we have chosen to use a switch. The reasons for this are that it is a simpler way to achieve the same purpose and also, we could argue that it is also more efficient, as we need less screen space and less items to achieve our goal.
* We have added a “saved tabs counter”. This is a text label that displays the number of tabs that we have saved in our app. This could be handy for the user, but it is also for us as developers as it was a quick and easy way to show that the tabs were being saved without having to debug each time. It was also possible thanks to the change above, that left a portion of the screen free.

Changes can be seen on the figure on the right.

## A screenshot of a phone Description automatically generatedSprint 2

User story:

“As a chords app for musicians user, I want to be able to see a list with the tabs that I have got saved in my library”

Higher value tasks:

* Create a recycler view to be able to display the saved tabs.
* Set up the recycler view to allow the user to interact with each individual tab.

Lower value tasks:

* Create an individual tab screen.

Development of sprint:

The high value tasks were completed successfully on addition to another large task that we hadn’t planned to execute just yet. When creating the view to display each tab on the main screen, we had all the uploaded tabs data stored in the upload activity itself, inside an array. The plan was to create the recycler view on the main screen and use the data we had in the array to test it, and leaving creating a local database for another sprint further ahead.

The issue was that sending the fetching the data from the upload activity was harder than expected so it was worth it to work on creating the database to store the data and then fetching the data from there into our main screen view directly.

We completed the creating of the database, although editing and deleting individual entries is not yet available.

We also changed the design of the main screen slightly by making the add floating button smaller and placing it on the right bottom corner of the screen. This will improve the overall usability of the recycler view and the overall user experience.

### Sprint 3

User story:

“As a chords app for musicians user, I want to be able to click on each tab of the list and see a detailed view of the tab”

Higher value tasks:

* Create an activity for individual tab visualization.

Development of sprint:

## Sprint 4

User Story:

“As a chords app for musicians user, I want to be able to interact with each tab that I have saved on my tabs screen”

Higher value tasks:

* Create an options menu for each tab on the list.
* Create a “delete” option for each tab.
* Create a “edit” option for each tab.

Development of sprint:

We spent much more time than expected choosing which type of options menu for each item of the list we were going to implement. After deciding we were going to use a three dots options button at the right side of each list item, we proceeded to implement this. Implementing a Popup Menu was simple enough but then we had to figurate out how to control the outcome from the Main Activity, because it is there where we have stored most of the data we want to modify. The solution was creating an interface on the recycler view adapter that we could access from the Main Activity.

After this, we created a function on the database helper object to delete a single tab from the list. Then we implemented this function on the delete option of the menu.

The edit option is more complex and time consuming, so we didn’t have time in this sprint. It will be our high value task for the next sprint.

# References

Agile Alliance. (2018, February 14). What Is Extreme Programming (XP)? Agile Alliance; Agile Alliance. <https://www.agilealliance.org/glossary/xp/>

Chordify . (n.d.). *Chordify Premium - Become a rockstar! - Chordify*. Chordify.net. Retrieved November 25, 2023, from https://chordify.net/premium

Google LLC. (n.d.-a). *Chordify - Instant Song chords - Apps on Google Play*. Play.google.com. Retrieved November 25, 2023, from https://play.google.com/store/apps/details?id=net.chordify.chordify&hl=en&gl=US

Google LLC. (n.d.-b). *SongbookPro - Apps on Google Play*. Play.google.com. Retrieved November 25, 2023, from https://play.google.com/store/apps/details?id=com.songbookpro.songbookpro&hl=en&gl=US

Google LLC. (n.d.-c). *Ultimate Guitar: Chords & Tabs - Apps on Google Play*. Play.google.com. Retrieved November 24, 2023, from https://play.google.com/store/apps/details?id=com.ultimateguitar.tabs

SongBookPro Systems Limited. (n.d.). *Download*. SongbookPro. Retrieved November 25, 2023, from <https://songbook-pro.com/download/>

OpenSongApp. (n.d.). *OpenSongApp - Current versions*. Www.opensongapp.com. Retrieved November 26, 2023, from <https://www.opensongapp.com/updates/latest-updates>

Glassdoor LLC. (2023, November 23). *Graduate Software Developer Salaries*. Glassdoor. <https://www.glassdoor.co.uk/Salaries/graduate-software-developer-salary-SRCH_KO0,27.htm#:~:text=The%20average%20salary%20for%20Graduate,from%20%C2%A3972%20%2D%20%C2%A33%2C487>.

Beck, K., & Andres, C. (2004). *Extreme Programming Explained : Embrace Change*. Addison Wesley.

Stellman, A., & Greene, J. (2015). *Learning agile*. O’Reilly.

Google LLC. (2019). *Google AdMob - Mobile App Monetization*. Google.com. <https://admob.google.com/home/>

Lyfe Marketing. (2021, February 23). *How Much Do Google Ads Cost? 2022 Price Guide (New Info)*. Www.youtube.com. https://www.youtube.com/watch?v=NAx0HlJqqJo

‌