Tune Talk
Final Report

By

Group: Team Melli
Course Title: SENG2021 T1 2021
Mentors: Fethi Rabhi, Decler Hague

CONTENTS

Overview 2
Project Purpose 2
Design Features 3
Forums 3
Recommendation System 3
Deliverable and Limitations 4
Successful Deliverable Items 4
Unsuccessful Items 4
Limitations 4
User Stories 5
Primary User Stories 5
Secondary User Stories 6
Sequence Diagram 7
User Interface 10
Project Components and Software Architecture 15
Frontend 16
Backend 16
External Data Sources: API 17
Alternative Architecture Consideration 19
Key Benefits and Achievement Designs 20
Conclusion and Appraisal of Work 21
Responsibilities 21
Issues and Problems 21
Opinions and Changes 22

OVERVIEW

Tune Talk is a web service that aims to cater to music enthusiasts. It allows users to make new song discoveries and talk about their favourite tunes. Tune Talk also intends on handling issues and the lack of features users complain about when using competitors' music platforms.

This report will discuss the purpose, functionality, design components and give an overview of the process and challenges the team faced in the duration of this project.

This project created by the Team Melli group who are a part of the SENG2021 course in Term 1 2021.

PROJECT PURPOSE

The main purpose of our system is to provide an alternative platform for access to music, discuss all things music and interact with other fans, as well as receive personalized music recommendations.

The two main problems that were addressed were:

- The lack of social networking on music apps and the lack of platforms made specifically for music discussions.
- The difficulty of discovering a wide array of new music based on music apps recommendation systems.

DESIGN FEATURES

The two main features that were the focus of this system were a forums discussion page for music, and a personalised music recommendation system.

- Forums - A forum page with posts that address certain music topics and groups, with the option to post, comment, share and interact with other users of the forums.
- Recommendation system - A recommendation system that focuses on data such as favourite artists, genres, songs as well as new music releases to facilitate the discovery of new music for users.

Forums

Tune Talk provides a forum space for users to share their thoughts and allow them to discuss and questions to the general users and music enthusiasts. Competitor platforms such as YouTube offers a pseudo forum sections to its users through the “Comments section”, however, the discussions are constrained under the space of the music video. It falls on the user to remember and navigate back their video of choice to access and interact with the comments again. Tune Talks brings the posts to the users at the forefront of their dashboard, offering convenience and allowing them to keep their focus on the discussions.

Recommendation System

Tune Talk also provides a personalized recommendation of music to each individual and aims to offer them a unique and customised experience. The web service monitors the categories of posts users interact with and suggests song playlists depending on that data. The search history of the user would also be taken into considerations when recommending new tunes. This aims to address the issues users face when using apps like Spotify where the individual receives generic music recommendations that are unrelated to their music tastes or receive suggestions which are exclusively based on only what the user hears, minimising their chances of discovering new tracks.

DELIVERABLE AND LIMITATIONS

There was a list of successful deliverables items, unsuccessful mentions and limitations that were implemented for the project. The challenges of implementing these are further discussed in the conclusion and appraisal of work.

Successful Deliverable Items

- User: Registration, login, authentication and interactions on Tune Talk
- Forums: Navigation, categorisation and updated interaction/view status
- Music recommendations: Search, tag filter and song results

Unsuccessful Items

- Forums: Post, comment, reply, like and react
- Music recommendations: Detailed song results, embedded video player and personalised song recommendations

Limitations

- Hosting the web server locally
- Unable to access the web service publicly (requires registrations of user)
- Examples songs recommendations limited to popular categories of music

USER STORIES

Primary User Stories

Story 1 – Forums

Feature	Make posts on the forum
Description	As a listener who enjoys analysing music So that I can talk with other people about music I want to make posts on the forum for everyone to see
Scenario	GIVEN I am on the 'Tune Talk' dashboard WHEN I click on "Forums" tab THEN I will be redirected to the "Forums" page WHEN I open up a "Popular post" THEN I should be taken to a discussion page which has title, author's post and comments WHEN I click on "Write comments" THEN a text input field should open up WHEN I write my comments AND click post THEN I should see a refreshed page which shows my post in the discussion page

Story 2 – New Music Recommendations

Feature	Recommend new songs
Description	As a user of 'Tune Talk' So that I can find new music to listen to I want to be shown a list of songs that I may be interested in
Scenario	GIVEN I visit Tune Talk Dashboard WHEN I click on 'Find new music' tab THEN a list new songs by new artists from similar genre is presented WHEN I click on a song from that list THEN The song begins to play

USER STORIES CONTINUED…

Story 3 – Personalised Music Recommendations

Feature	Recommend similar songs
Description	Be recommended similar songs by previously heard artists As a user of 'Tune Talk' So that I can find new music to listen to I want to be shown a list of songs by new artists I may like
Scenario	GIVEN I visit Tune Talk Dashboard WHEN I click on 'Find similar music' tab THEN I will be shown a list of music based on my taste and recent post engagement WHEN I click on a song from that list THEN The song begins to play

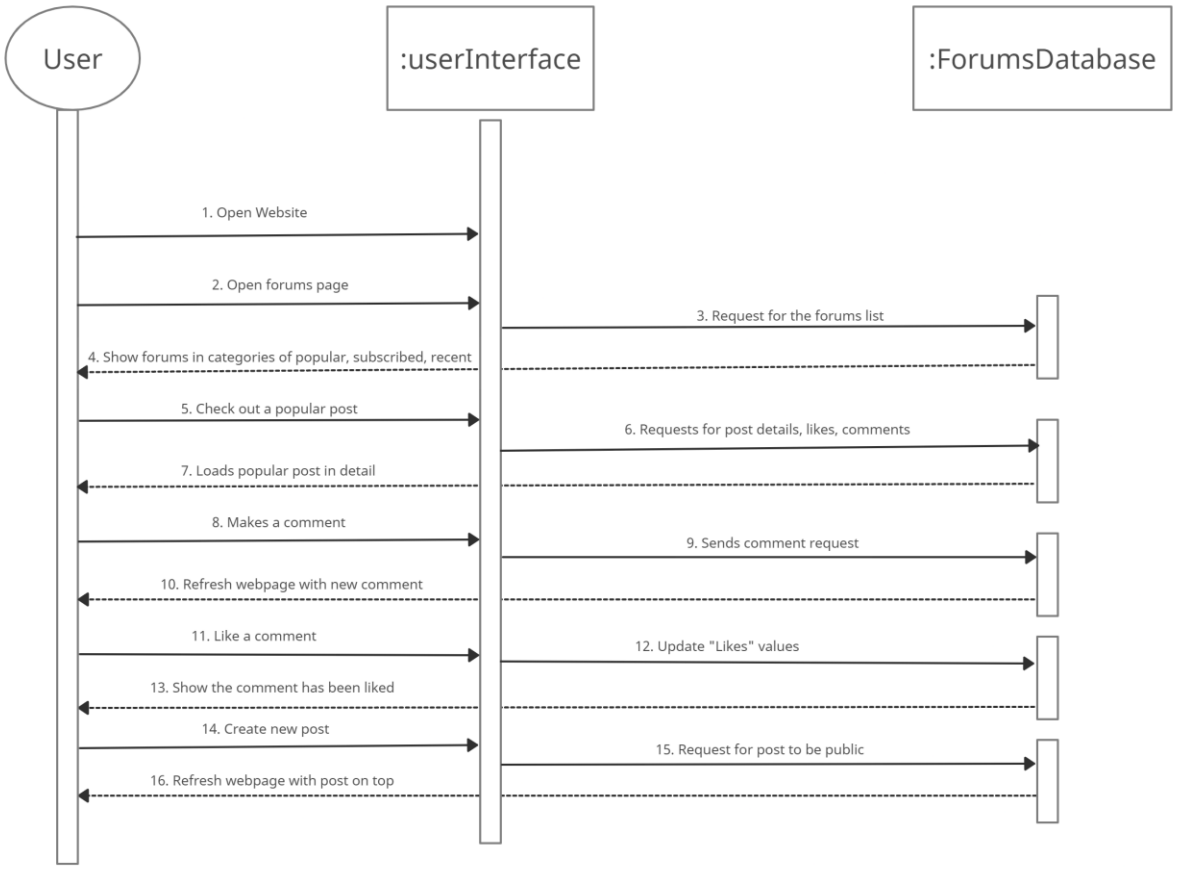
Secondary User Stories

Story 4 – Song Details

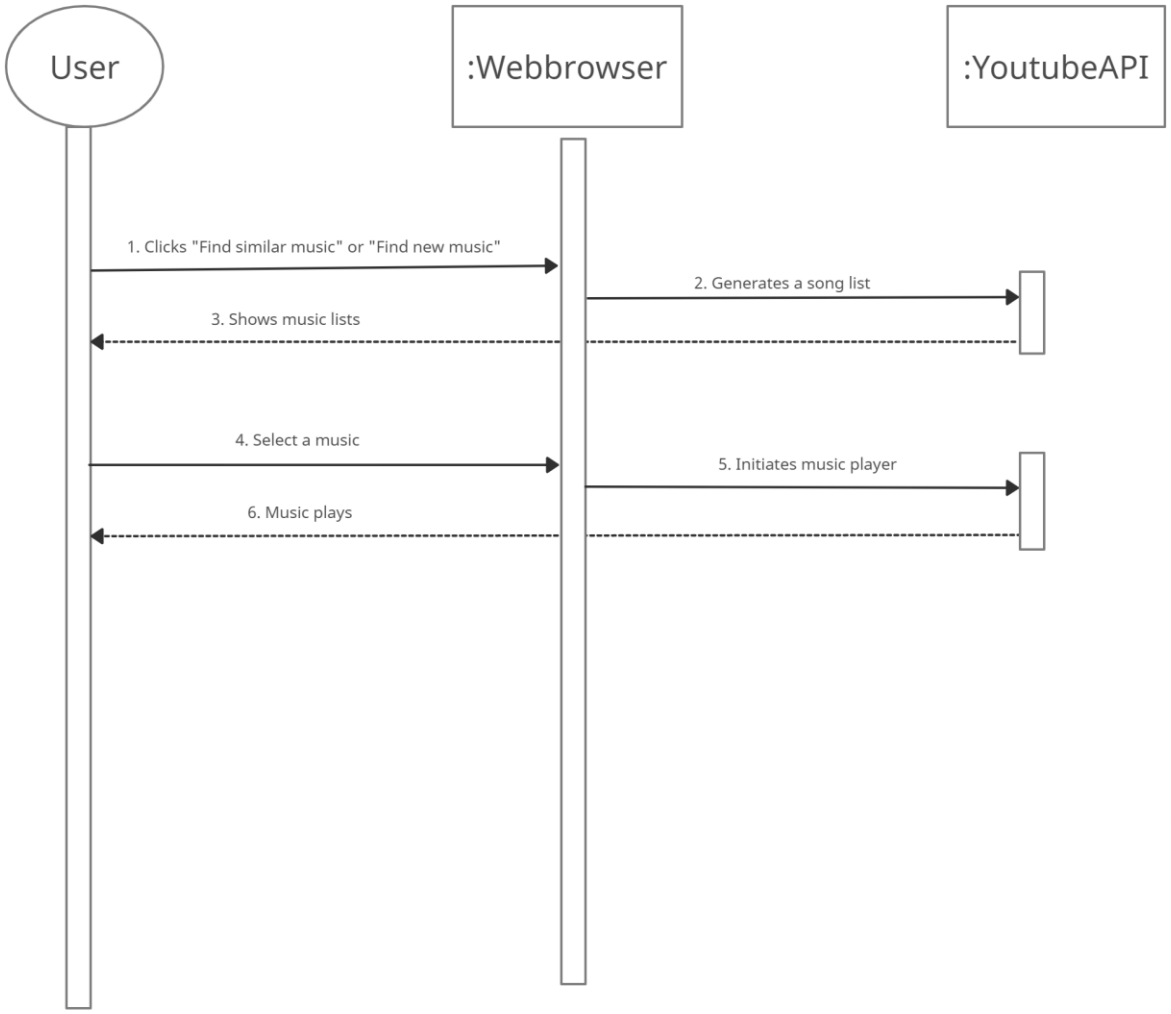
Feature	View song details
Description	As a user of 'Tune Talk' So that I can find information on a song I want to be view a song detail when I search for it
Scenario	GIVEN I visit Tune Talk Dashboard WHEN I click on the song search bar THEN I will be able to type in a song I want to know more about WHEN I click on the 'Search' icon THEN I will be shown a list of song results that matched my search WHEN I select a song THEN I will see a new tab open with the song details such as: - Name, album - Artist - Release date - Lyrics

SEQUENCE DIAGRAM

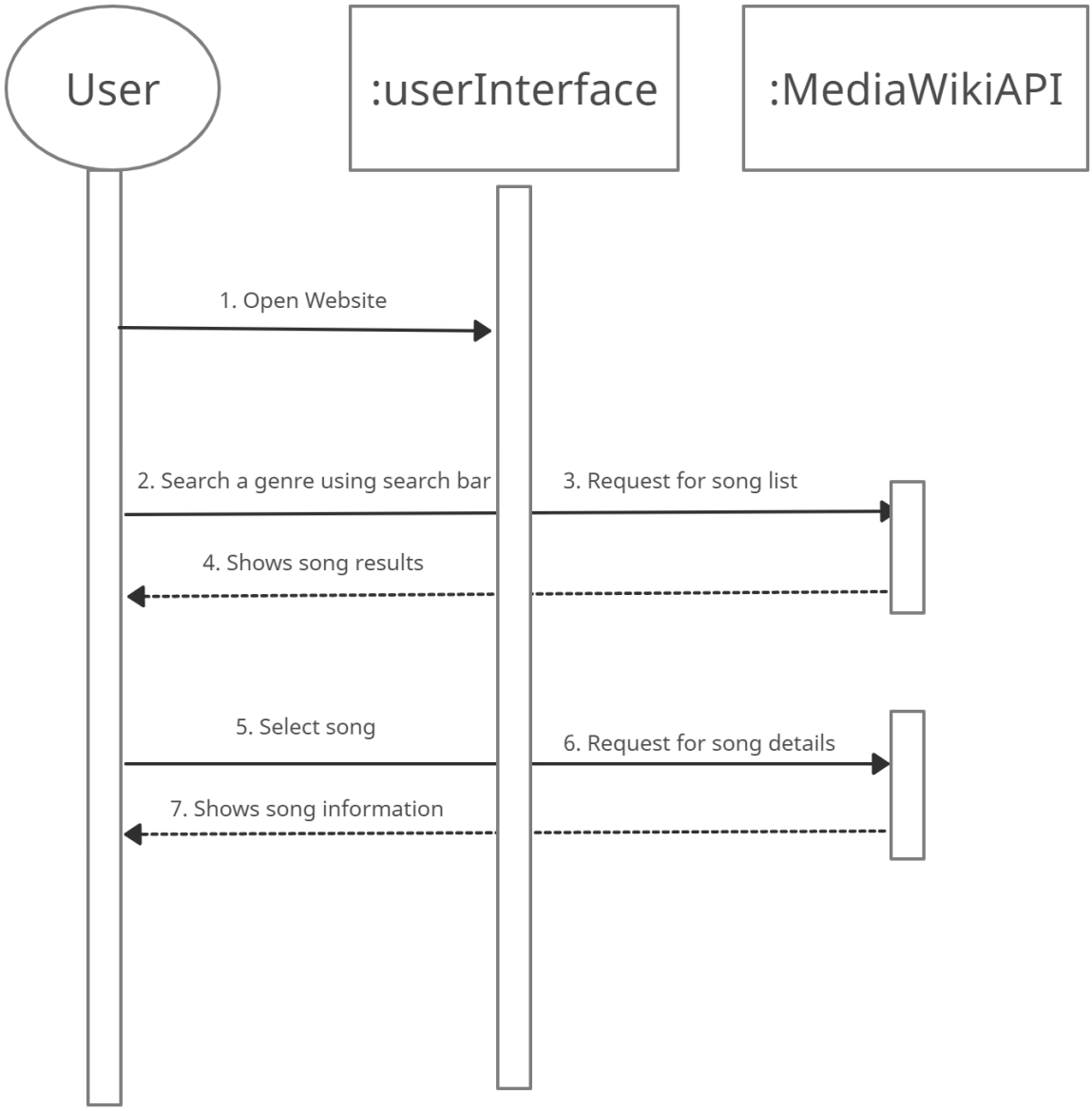
Story 1:



Story 2 and Story 3:

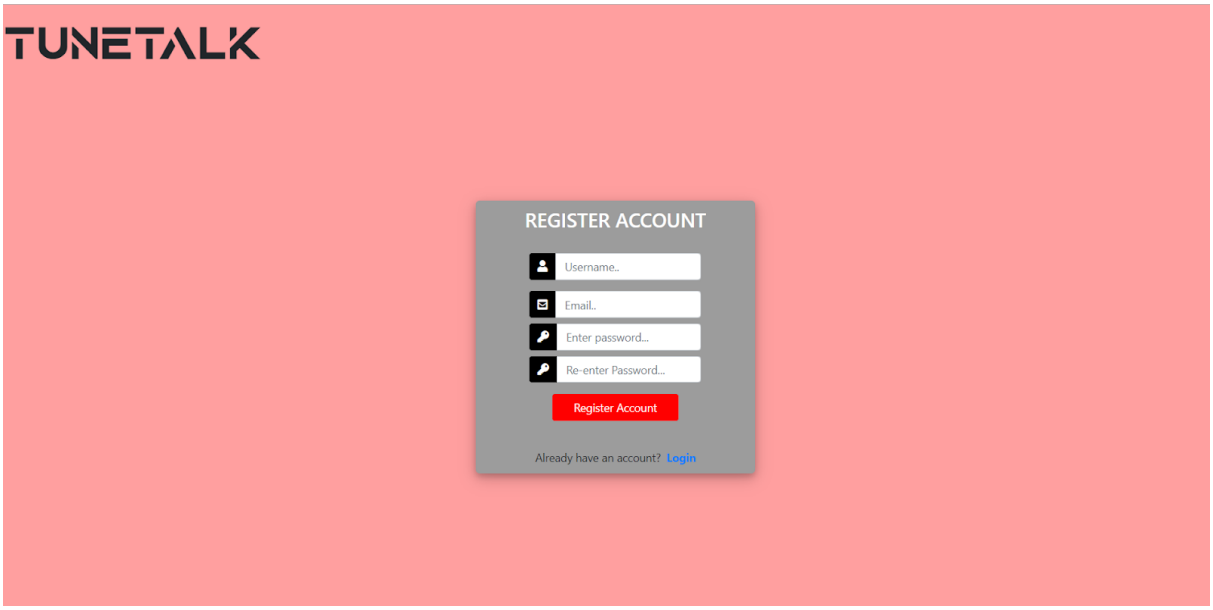


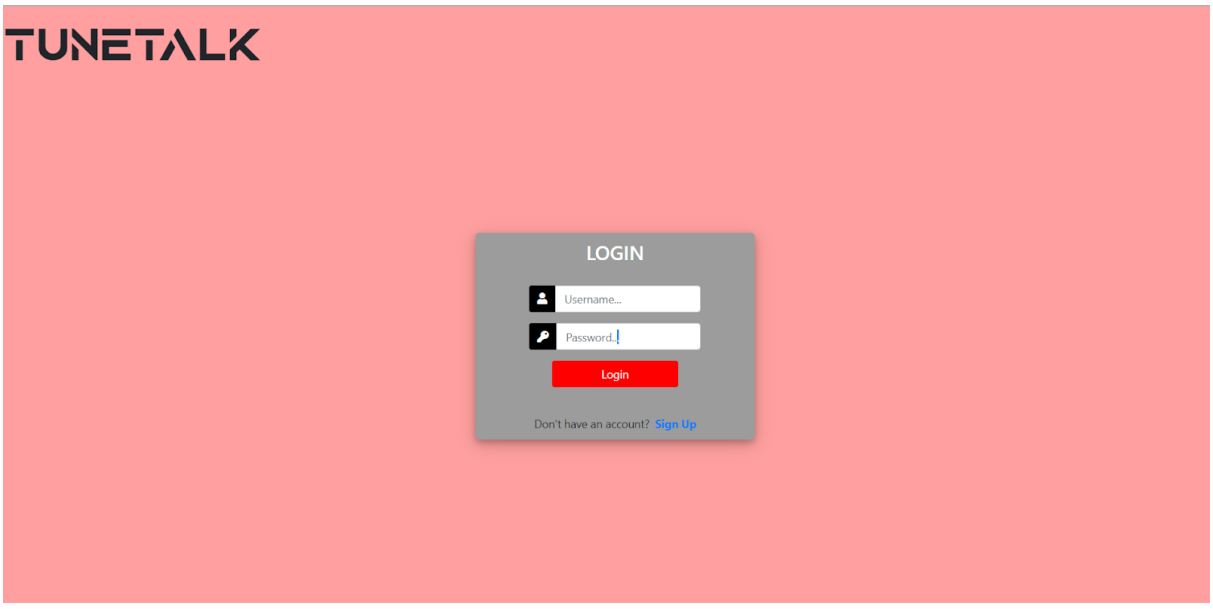
Story 4:


User
:userInterface
:MediaWikiAPI
1. Open Website
2. Search a genre using search bar
3. Request for song list
4. Shows song results
5. Select song
6. Request for song details
7. Shows song information

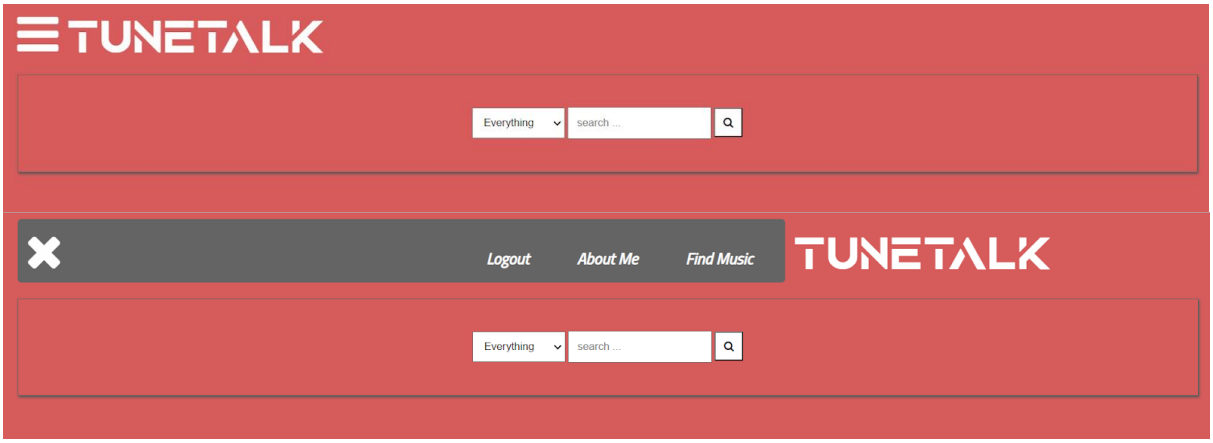
USER INTERFACE

The registration and login page:

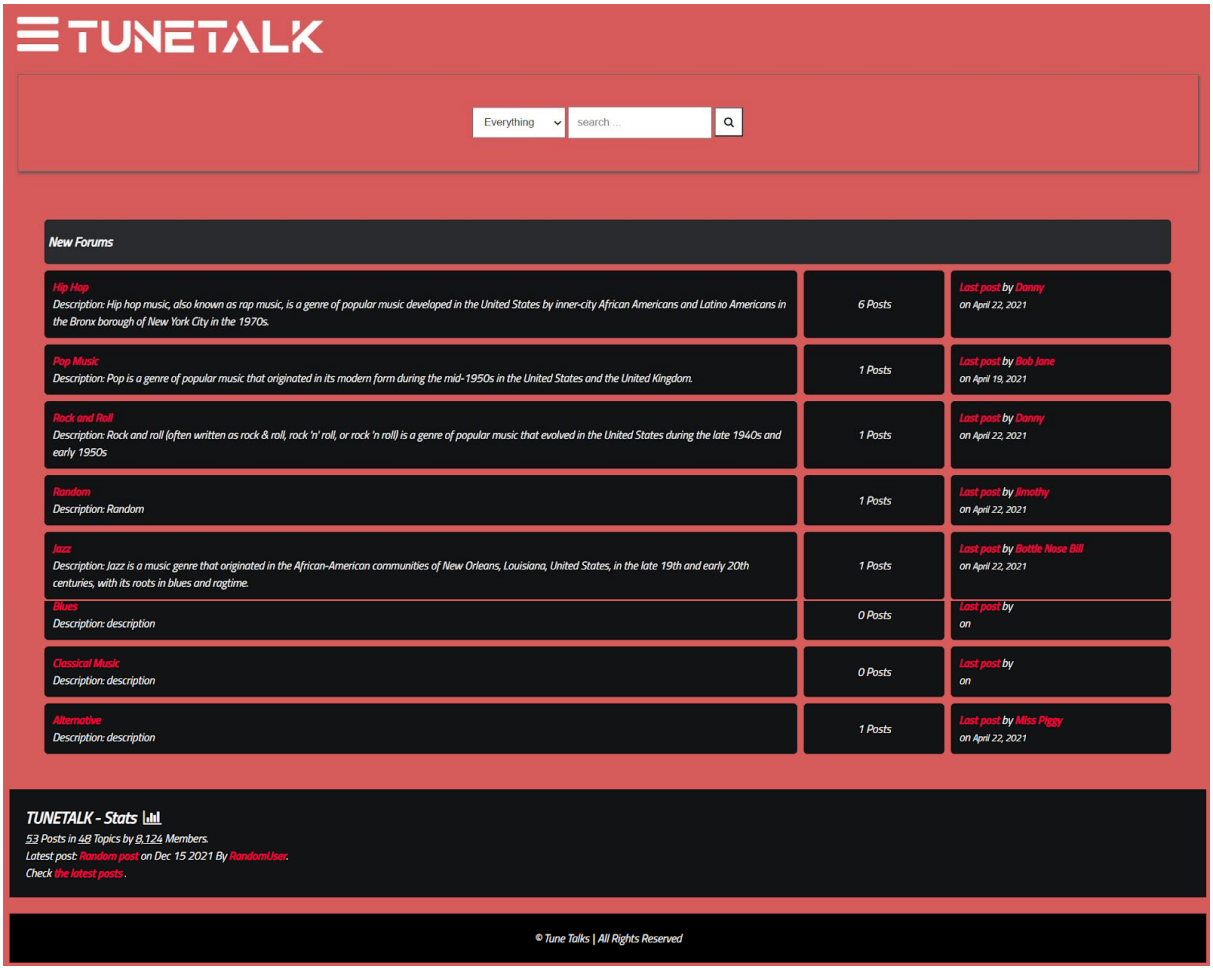




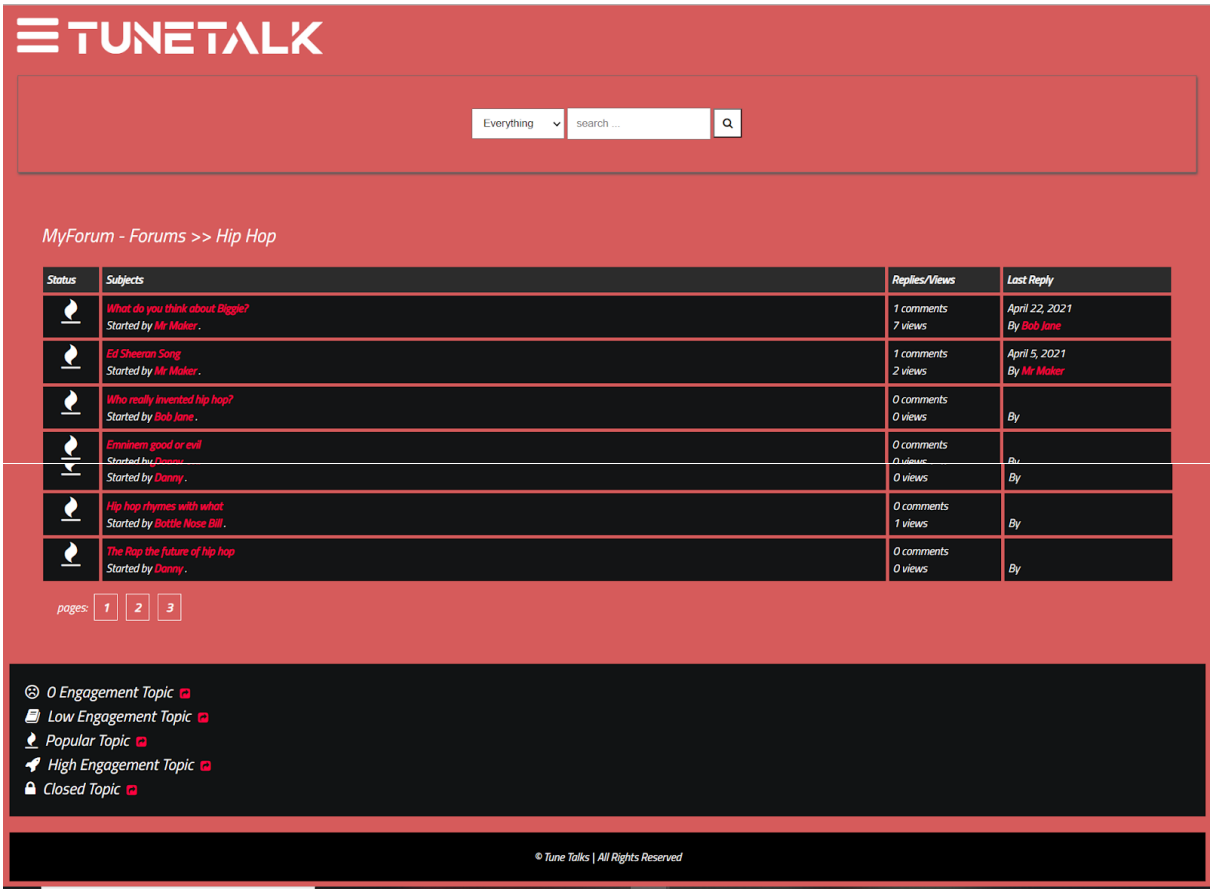
Navigation bar and search bar:



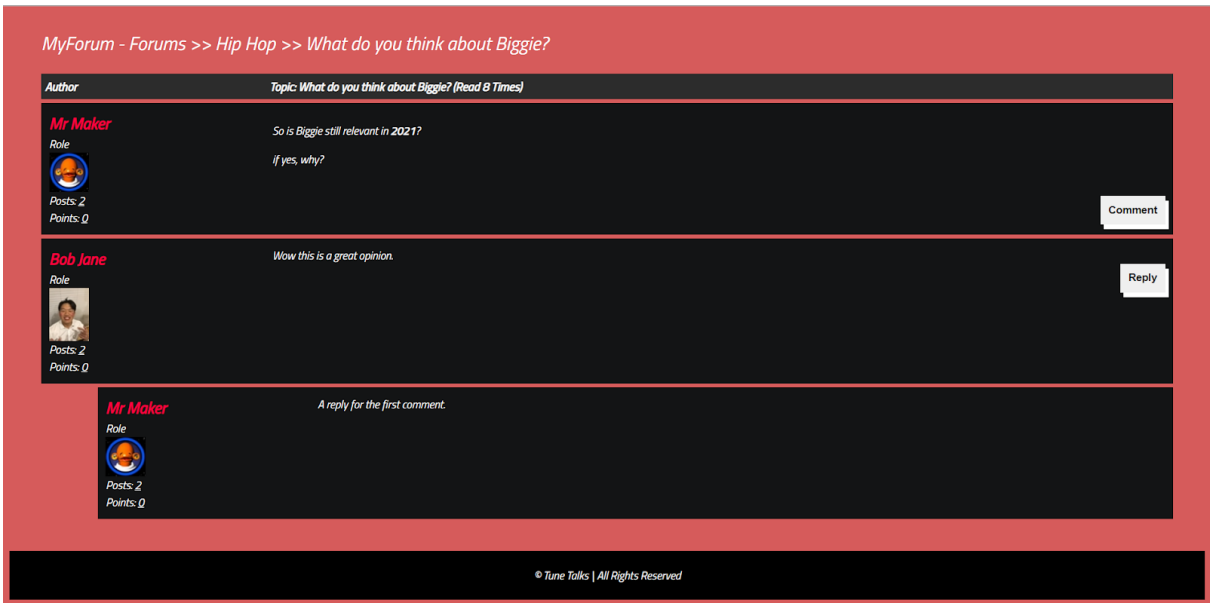
Forums main page:



Posts listings:



Detailed post page:



MyForum - Forums >> Hip Hop >> What do you think about Biggie?

Author	Topic: What do you think about Biggie? (Read 8 Times)
Mr Maker Role Posts: 2 Points: 0	So is Biggie still relevant in 2021? if yes, why?

Comment

comment here ...

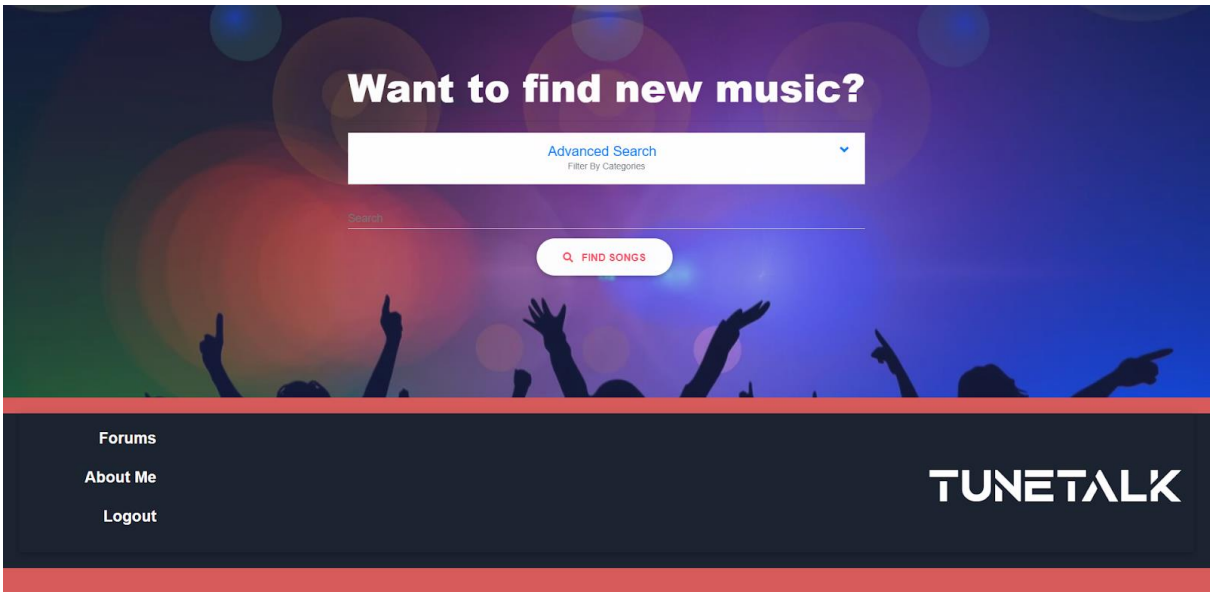
submit

Bob Jane Role Posts: 2 Points: 0	Wow this is a great opinion.

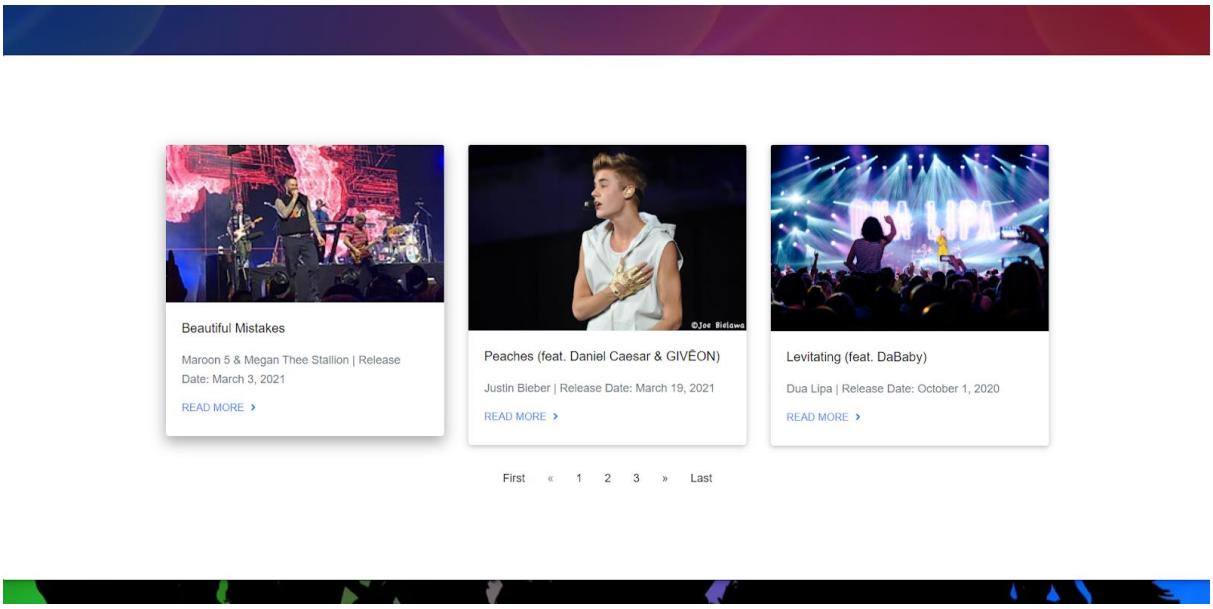
Reply

reply here ...

Music recommendation main page:



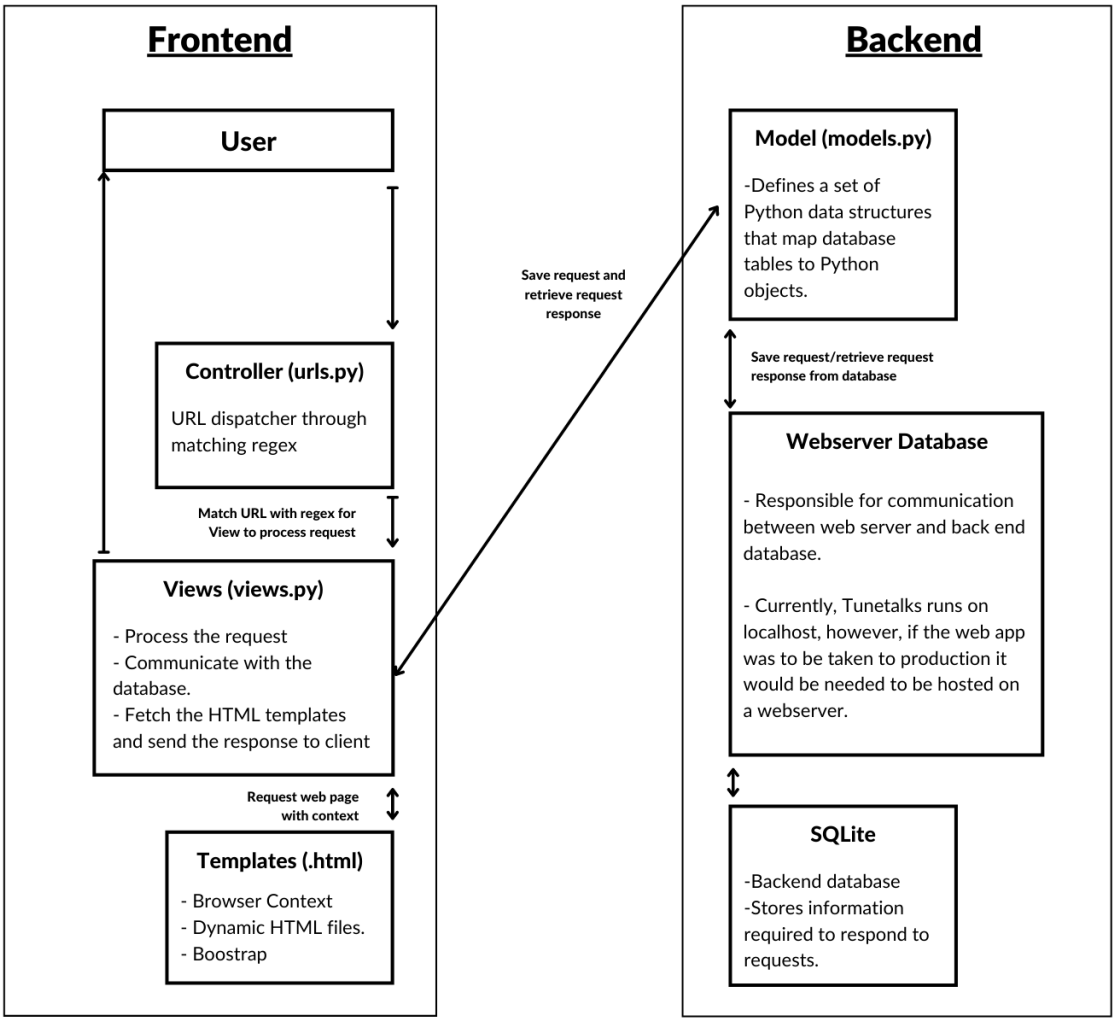
Search result page:



PROJECT COMPONENTS AND SOFTWARE ARCHITECTURE

The frontend and backend software components have been altered to utilise the pre-existing skills of multiple members in the group. This change had to be made as there was a time limited to learn a new language and a new framework.

The UML Diagram:



Frontend

The front end utilises the Django framework. It is a high-level framework that allows for development in a short period of time using Python. Django includes pre-existing Web pages that are already setup, allowing the project to save time through bypassing the Web development stage.

Django offers accessibility as resources and tutorials already exists in regards to the functionalities intended for Tune Talk. This allowed the group to quickly pick up the framework and start customising the forums features and the recommended music features.

There was overall advantages and disadvantages of Django framework.

Advantages

Django includes packages that deal with administration and registration of users, authentication and API connections, this is favourable as Tune Talks deals with multiple users and utilises the API for the functionality of music recommendations.

Allows members to work on separate components remotely, this is useful as one member can work on implementing the “Recommended Music” whilst another can work on “Forums” functionalities.

Provides security and authentications which protects sensitive information being handled in the server when the user login and registers.

Disadvantages

The monolith nature of Django means members have to learn about the sets of predefined variables before being able to utilise the framework, this however provides consistency in the project between all members’ coding.

The files structures are rigid, this means creators cannot alter anything or customise it for their own suitability, however learning curve of the structure system is easy.

Backend

Django comes with an ORM that lets you manipulate your data source with ease, forms (an HTML independent implementation) to process user input and validate data and signals, and an implementation of the observer pattern. Plus a tons of use-case specific nifty little tools. For the rest of the backend work Django doesn't help with, you just use regular Python. Business logic is a pretty broad term.

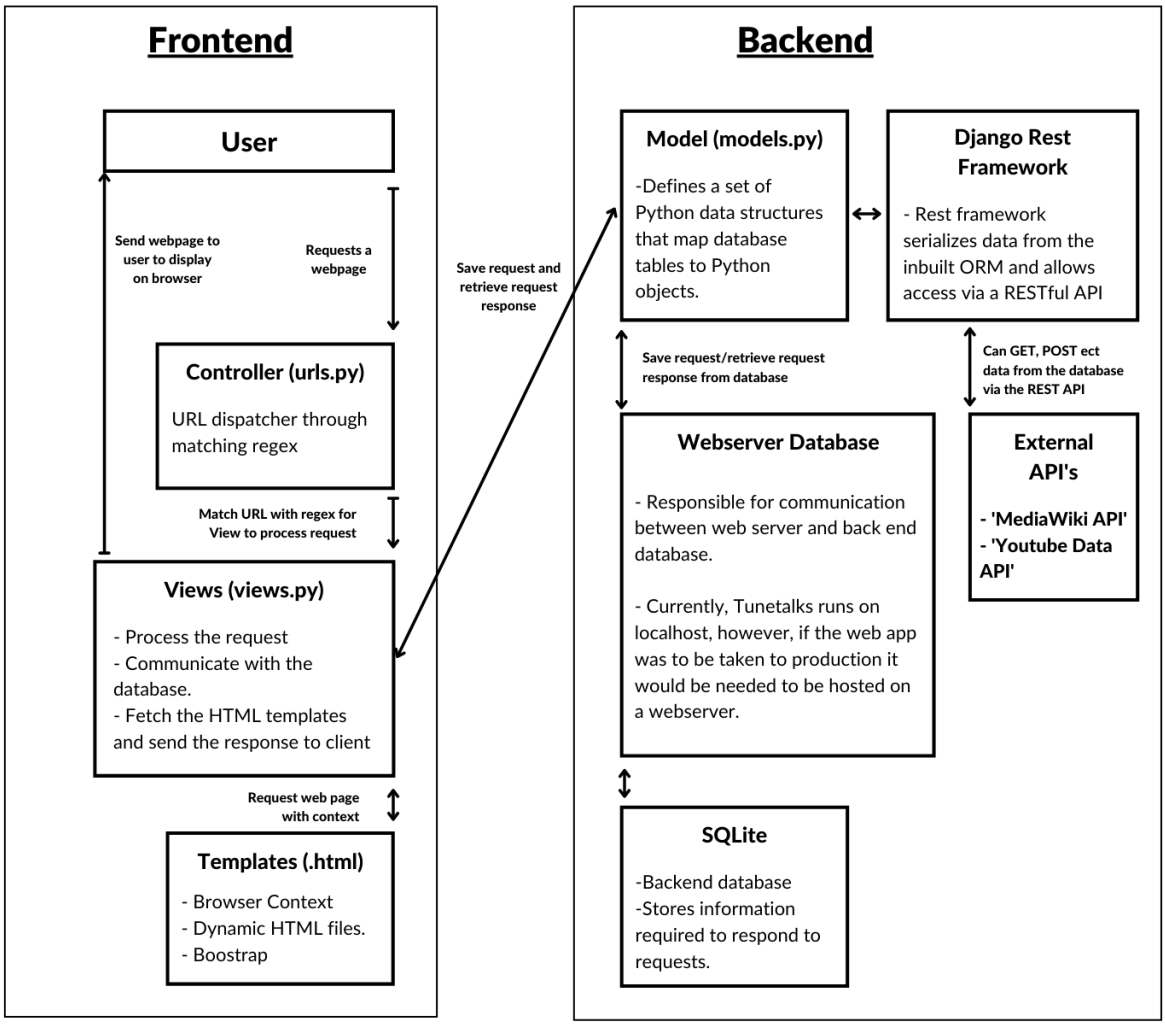
External Data Sources: API

Although the project at its current stage has not utilised any API, these were the intended one the project would utilise and UML diagram that would integrate these API into the system.

The main external data source our system will be accessing is Wikipedia through the 'MediaWiki' API. Data is received from this API when users search for songs or are linked to the music detail page. This is the most relevant API to our system.

Firstly, our choice of the YouTube Data API was due to its extensive use in the music area. The advantages of it include being able to search by specific fields such as artists, publication dates, genres and location. Users can search not only for specific videos but also playlists and artist channels, and search by language. The reason the YouTube Data API was chosen over the Spotify or Apple music API is because our system will be a stand-alone platform rather than an add-on to Spotify or Apple Music. Furthermore, unlike the Spotify and Apple Music API, YouTube Data API and MediaWiki offer free solutions for the project.

The UML Diagram with API integration:



ALTERNATIVE ARCHITECTURE CONSIDERATION

These were the alternative components considered to build 'Tune Talk'. The benefits and drawback were compared and the reason for changes were reflected upon.

dotCMS	
Advantage	- High flexibility and customisable: Able to change to fit the specific music themed functionalities - Easy to learn interface
Disadvantage	- Security concerns: The site is more vulnerable to cyber-attacks as parts of the code are extracted from an open-source platform - Site speed: Excess generic code and plugins associated can potentially increase loading time of the site, resulting in a worse user experience - The project members are not familiar with the Java language used to work with dotCMS

Nust.js	
Advantage	- Extremely light weight, and most commonly used for single page applications - A minimalist framework for server rendered applications
Disadvantage	- Does not meet the requirements of Tune Talk as it deals with multiple pages and functionalities

Angular	
Advantage	- Angular supports MediaWiki API and Youtube Data API integrations, which is the external data sources the project utilises - Greater community support allowing access to help and resources
Disadvantage	- Complex, beginners needs significant time to understand development instrument - The search engine crawlers is poor

KEY BENEFITS AND ACHIEVEMENT DESIGNS

The main feedback of our design questioned why users would be interested in joining 'Tune Talk' to listen and discuss music rather than commenting on available platforms like YouTube or YouTube Music/Spotify which are already integrated into their everyday routine. YouTube offers a general form of entertainment which recommends its users comedians, bloggers, commenters, and etc. The trending and recommendations tabs reflect this focus, hence the work is on the music listener to shift through the various categories of videos to search for music they are interested in. Although this issue can be bypassed through the usage of YouTube Music, this brings forth the problem of the lack of social networking as mentioned in the user stories, as YouTube Music does not allow listeners to comment on the song videos. Note, this issue is also prevalent with using Spotify.

In addition, music discussions on YouTube are restricted and limited to a video by video basis, users would have to seek out these discussions manually and search for specific videos to read the comments section about a song or an artist. Our platform however, brings all of the discussion threads to one place and makes it much more convenient to navigate based on certain criteria such as a song, artist, album or even genre. It is much more interactive than a comments section and it is much easier to access and navigate, making it much more valuable in terms of business appeal over other platforms with comments features.

For music recommendations, one of the biggest complaints about Spotify is that their music recommendations system is very generic and often lists songs very similar to music that users have already listened to. Our system brings a more personalised recommendation system and takes into account a wider variety of factors when making recommendations, in order to make sure users have an easier time discovering new music and expanding their music database.

CONCLUSION AND APPRAISAL OF WORK

Responsibilities

When organising our tasks, we met over discord and zoom calls to agree on our roles for each deliverable and the project as a whole, as well as to assign responsibilities. We split up work based on our strengths and levels of experience. For example, Arsham and Gong were in the role of programmers and developers who handle the coding. Kasra was in charge of report writing, Varun did our research and they were the two main speakers in the presentations. Afia helped both groups and also worked primarily on the diagrams and use cases as well as our user stories. In terms of planning and decisions for the project, we initially brainstormed and discussed our ideas together in order to come to a final decision. For our major software decisions and project ideas we all talked and voted on the ideas that most of us agreed with.

Issues and Problems

As with any group project, we faced a few challenges and problems. Our biggest problem was communication as we were not able to constantly be on the same page about every aspect of our work and this sometimes set us back and lowered our productivity. Although we made efforts to meet often we were unable to meet as much as we wanted to due to scheduling conflicts and different time zones. A part of this problem was also due to the online nature of the course as we believe it would be different if it was in person, however this is obviously a special circumstance that we control. Another issue we faced was that in some parts of our project, we took too much time before making our major decisions, which made our subsequent work feel rushed and lowered the quality as opposed to a situation where we had more time.

In some of our technical aspects we faced a few issues, for example some of our code was not functioning on the backend of our server, however after working on this we managed to fix most of it. We also had a few issues with our diagrams as we were not exactly sure what was expected, and had to re do a few of them. However once we received clarification we were able to complete these on time.

Opinions and Changes

As a group we feel that our project as a whole was a great idea and that we worked together as best as we all could. We feel as if the end result was satisfactory but could have definitely been much improved. In terms of changes, we would probably go about our planning stage differently now as we realised we took too long to brainstorm. We would focus more on getting our work started earlier in the term and making changes along the way if necessary, to avoid falling behind.

In terms of the technical aspect, we had initially planned to use a CMS system to build our web application. This would have had various advantages such as high customizability and the integration of several different modules and APIs. Ideally we would have liked to have this system built on dotCMS or a similar platform as it would have resulted in a much better final product that was more aligned with the design ideas that we had envisioned for this project. Due to time constraints we saw that we would not be able to learn and deploy a CMS based system in the time remaining to complete our project, and therefore decided to use Django to build our site. Obviously it was not our first choice as it did not allow us to implement all of the software design that we had come up with and it did not let us implement our best features to their full capabilities. However, there were a few advantages to this platform. Firstly it was much easier to learn and get working on, so this saved us a lot of time and allowed us to successfully complete our web application on time for our final due date. It was also convenient to work with and it used python which was something we did not need to completely learn as we had experience with it before, making our work easier.

Another change that we would have made after evaluating our work this term is to focus on a smaller amount of features initially rather than trying to work on many. We initially had 4-5 different unique services/features in mind for our project but this was not realistically achievable in the time-frame of our project. Eventually we narrowed our focus down to the 2 main features of a forums page and a music recommendations system, however this was more than halfway through the term. If we had focused on these much earlier in the term, our design process as well as our coding and implementation would have gone much smoother. This would have allowed us to build on the 2 main features and make them better versions of what we were actually able to implement in our given time frame.

Our time management was good as a whole, as we set deadlines and did our best to keep up with our work through the term, however our allocation of time could have been better. We spent a lot of time on brainstorming and planning, as well as design early in the term, and we did not make our design choices in a timely manner. We spent more time than necessary on the reports and user stories whereas if we were to do it all over again, we would spend more time on coding, testing and adding to our actual web application.