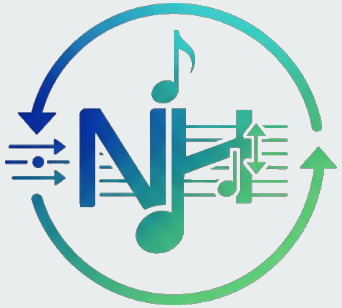


NewHarmonics

Project for the Laboratory of Advanced Programming course

Luca Cornici

Onorio Iacobelli

Alessandro Rocchi

Overview

New Harmonics is a music sharing web app where users can register either as artists or listeners.

Artists can upload songs with basic metadata (title, album, genre), while all users can browse and play music.

The system includes a basic dashboard that displays popular songs and tracks from artists the current user is following.

Users can also download or like songs in order to save them and get notifications regarding the latest uploads of their followings.

User Stories

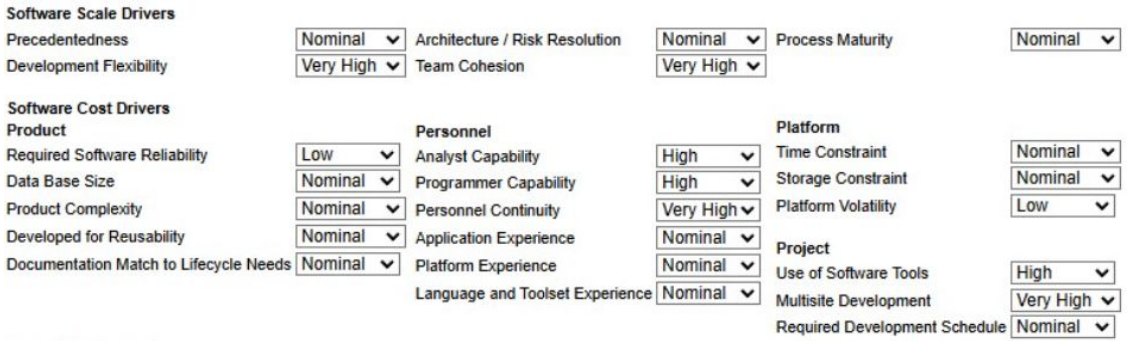
1. As an unregistered user, I want to create an account, so that I can start using the application
2. As a registered user, I want to edit my profile (name, email, password), so that it reflects my preferences.
3. As a registered user, I want to delete my account, so that I can permanently remove my profile and songs.
4. As a registered user who is logged out, I want to log in, so that I can access all the features.
5. As a registered logged-in user, I want to log out, so that I can leave the application securely.
6. As a user, I want to view the Home Page, so that I can explore the newest and most popular uploaded songs.
7. As a registered user, I want to follow an artist, so that I can quickly access their profiles and music.
8. As a registered user, I want to unfollow an artist, so that I remove them from my following list
9. As a registered user, I want to open my Following Page, so that I can see the artists I follow.
10. As a registered user, I want to open my Feed Page, so that I can see the recent uploads from the artists I follow.
11. As a registered user, I want to get notified when someone I follow uploads a new song, so that I can listen right away.

User Stories

1. As an artist, I want to upload a song or podcast, so that others can listen to my content.
2. As an artist, I want to see the list of songs I have uploaded, so that I can manage my content.
3. As an artist, I want to delete one of my uploaded songs, so that it is no longer available to others.
4. As an artist, I want to update the details of a song (title, genre, album, cover, audio file), so that the information stays accurate.
5. As a user, I want to listen to a song available on the platform through the player, so that I can enjoy music from different artists.
6. As a registered user, I want to add a song to my favorites, so that I can easily find it later.
7. As a registered user, I want to see the list of my liked songs, so that I can quickly access my favorite music.
8. As a user, I want to download a song, so that I can keep it stored in my system.
9. As a user, I want to search for songs by title, artist, or genre, so that I can find the music I want.
10. As a registered user, I want to access my Profile Page, so that I can see my details, favorite songs and followed artists.
11. As a user, I want to visit an artist's Profile Page, so that I can see their uploaded music.

Effort Estimation

- Unadjusted Function Points: 96
- Language: Java



Estimated results:

- Estimated hours = 125
- Schedule = 6.4 Months
- Total Equivalent Size = 5088 SLOC
- Effort = 6.2 Person-months

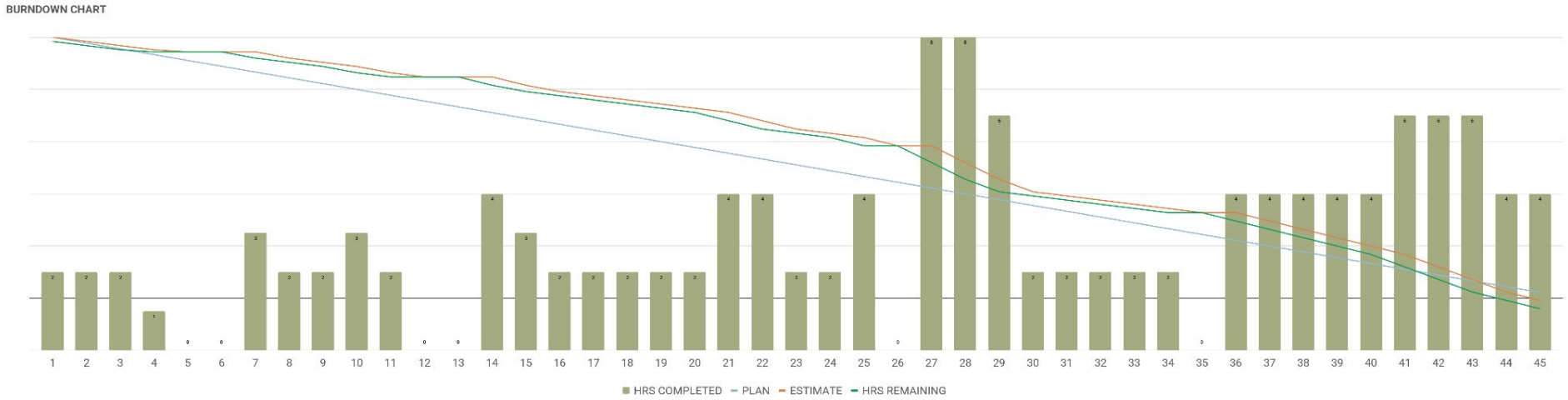
Actual results:

- Total hours = 129
- Total days = 45
- Total LOC = 8750

Source: https://softwarecost.org/tools/COCOMO/

# No.	Module	Function Name	Description	Type	DET	RET/FTR	Complexity	FP	Remarks
1	User Management	User	Model stored to represent user accounts (artists/listeners).	ILF	7	1	Low	7	Users table (H2). DETs: ID, Email, Pass, Role, etc.
2	Song Management	Song	Model stored to represent song metadata and files.	ILF	13	1	Low	7	Songs collection (Mongo). DETs: _id, title, artistId, fileUrl, likedBy, etc.
3	Follow	Follows	Model stored to represent the follow relationship.	ILF	2	1	Low	7	USER_FOLLOWS table (H2). DETs: follower_id, artist_id.
4	Notification	Notification	Model stored to represent user notifications.	ILF	11	1	Low	7	NOTIFICATIONS table (H2).
5	User Management	Register	(US 1) As an unregistered user, I want to create an account...	EI	4	1	Low	3	Modifies User ILF. DETs: username, email, pass, role.
6	User Management	Edit Profile	(US 2) As a registered user, I want to edit my profile...	EI	4	1	Low	3	Modifies User ILF. DETs: user_id, name, email, pass.
7	User Management	Delete Account	(US 3) As a registered user, I want to delete my account...	EI	1	3	Low	3	Modifies User, User Follows, Song ILFs (via RabbitMQ).
8	User Management	Login	(US 4) As a registered user... I want to log in...	EI	2	1	Low	3	Reads User ILF to validate. DETs: email, pass.
9	User Management	Logout	(US 5) As a registered logged-in user, I want to log out...	EI	1	1	Low	3	Modifies user session/token status.
10	Discovery	View Home Page	(US 6) ...explore the newest and most popular uploaded songs.	EO	3	1	Low	4	Reads Song ILF (via Elastic). Data is processed.
11	Follow	Follow Artist	(US 7) As a registered user, I want to follow an artist...	EI	2	1	Low	3	Modifies User Follows ILF.
12	Follow	Unfollow Artist	(US 8) As a registered user, I want to unfollow an artist...	EI	2	1	Low	3	Modifies User Follows ILF.
13	Follow	View Following Page	(US 9) ...see the artists I follow.	EQ	3	2	Low	3	Reads User Follows and User ILFs.
14	Feed	View Feed Page	(US 10) ...see the recent uploads from the artists I follow.	EO	4	2	Low	4	Reads User Follows & Song ILFs. Data is processed.
15	Notification	Get Notifications	(US 11) ...get notified when someone I follow uploads...	EQ	5	1	Low	3	Reads Notification ILF.
16	Song Management	Upload Song	(US 12) As an artist, I want to upload a song...	EI	6	1	Low	3	Modifies Song ILF. DETs: title, genre, album, files...
17	Song Management	View Uploaded Songs	(US 13) ...see the list of songs I have uploaded.	EQ	5	1	Low	3	Reads Song ILF.
18	Song Management	Delete Song	(US 14) As an artist, I want to delete one of my... songs.	EI	1	1	Low	3	Modifies Song ILF (and MinIO file).
19	Song Management	Update Song Details	(US 15) As an artist, I want to update the details of a song...	EI	6	1	Low	3	Modifies Song ILF.
20	Song Management	Get Song Details	(US 16) ...listen to a song available on the platform...	EQ	6	1	Low	3	Reads Song ILF. (Fetches data for player).
21	Song Management	Like Song	(US 17) As a registered user, I want to add a song to my favorites...	EI	2	1	Low	3	Modifies Song ILF (adds user to likedBy array).
22	Song Management	View Liked Songs	(US 18) ...see the list of my liked songs.	EQ	5	1	Low	3	Reads Song ILF (queries likedBy array).
23	Song Management	Download Song	(US 19) As a user, I want to download a song...	EO	2	1	Low	4	Reads Song ILF (gets file from MinIO).
24	Discovery	Search Songs	(US 20) ...search for songs by title, artist, or genre.	EQ	6	1	Low	3	Reads Song ILF (via Elastic).
25	User Management	View Profile Page	(US 21, 22) ...access my Profile Page / visit an artist's Profile...	EO	6	3	Medium	5	Reads User, Song, User Follows ILFs. Aggregates data.
			Total Unadjusted Function Points					96	

Burndown Chart

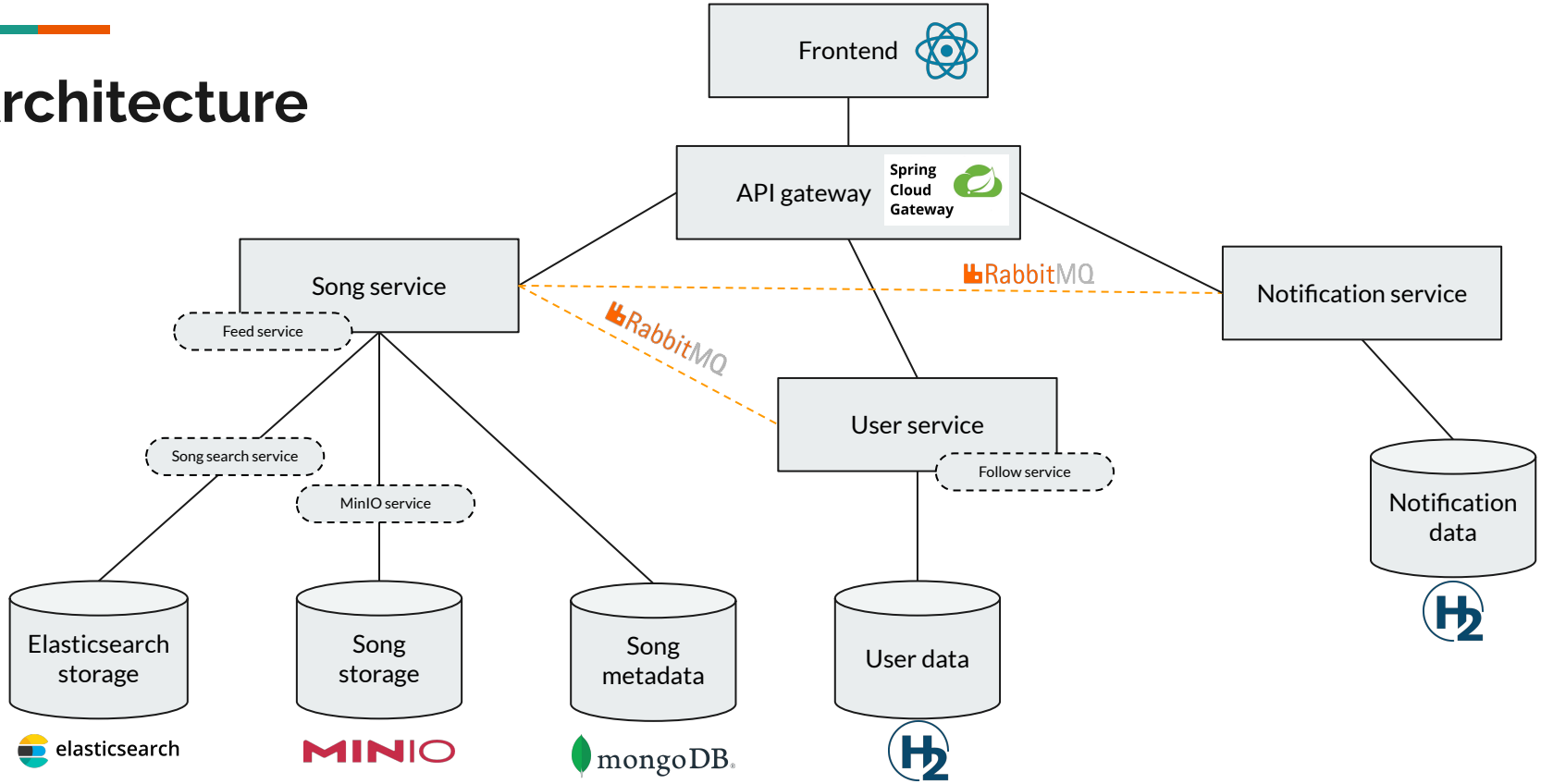


SCRUM Sprints

The workload has been divided in 5 sprints:

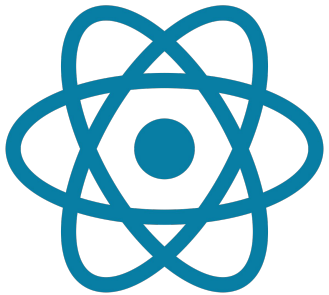
1. **Project definition** - initial brainstorming and planning
2. **Project setup** - definition of user stories and wireframes and effort estimation
3. **Core functionalities** - implementation of backend and frontend for song management and user service
4. **Advanced functionalities** - implementation of following, liking and feed, alongside the improved search and upload system and the notification system
5. **Documentation** - final revision and finalization of the documentation

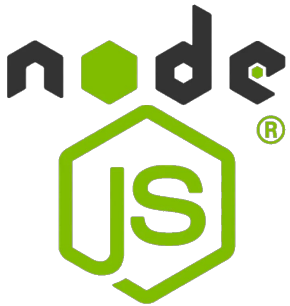
Architecture



Frontend

The frontend was developed using React and Node.js. A local server is hosted on port 3000 and requests are sent to the API gateway which forwards them to the related microservice.





API Gateway

The API gateway was realized with Spring Cloud Gateway. On top of handling routing, it also manages security, authentication (through JWT - JSON Web Tokens) and CORS.

Song Service

This service manages storage of metadata, upload and download of songs.

Song metadata is stored in a MongoDB database, while audio files and cover images are stored in the MinIO storage platform.

Feed Service and Song Search Service

The feed service is used to retrieve the feed of the current user, i.e., the page displaying the most recent songs uploaded by their followed artists.

The song search service handles the search of songs. Elasticsearch is used for indexing and querying, offering more features compared to classic DB querying.

User Service and Follow Service

The user manages user accounts and JWT generation. Accounts are saved in an H2 database.

The follow service handles the logic related to the following of user accounts. Data is saved in the same H2 db.

Notification Service

This service sends notifications to users to notify them of their followings' newest uploads.

Notifications are stored in the H2 db and RabbitMQ is used to handle the communication with the song service when a new song is uploaded.

RabbitMQ is also used to handle the removal of songs when the artist that uploaded them is deleted.

Demo

We can demo of the app by connecting to http://localhost:3000