{EPITECH.}

AudioWire

Hugo Defrance, Huai Xie, Stéphane Rose

Guillaume Feraud, Augustin Pasquini,

Vivien Meilhac, Maximilien Brunelle

About the document

The AudioWire project will lead to the realization of an audio player with an artificial intelligence. It will allow, for example, the user to create personalized playlists depending on the user's mood. Moreover, the player will include social functionalities. This document is going to present the realization of this project in details.

In the first part, we are going to explain what an EIP is and what is its objective. Then, we will detail the different functionalities expected at the end of the project. We will also study the economic context where the AudioWire project is and its targeted population.

Moreover, we will detail the different parts of the program before developing the different parts of the project itself. In this part, we will pay a particular attention to the technical architecture of the project and the existing components on the market. We will then develop the security aspects of the project and its sensible points.

Furthermore, we will study the different tools and constraints linked to the realization of this project before comparing the different kind of databases available on the market. We will see what database will be chosen for the project and why.

Finally, we will see the different functionalities that will have to be tested and validated before the end of the project. To finish, we will develop the steps of realization of the project.

Document description:

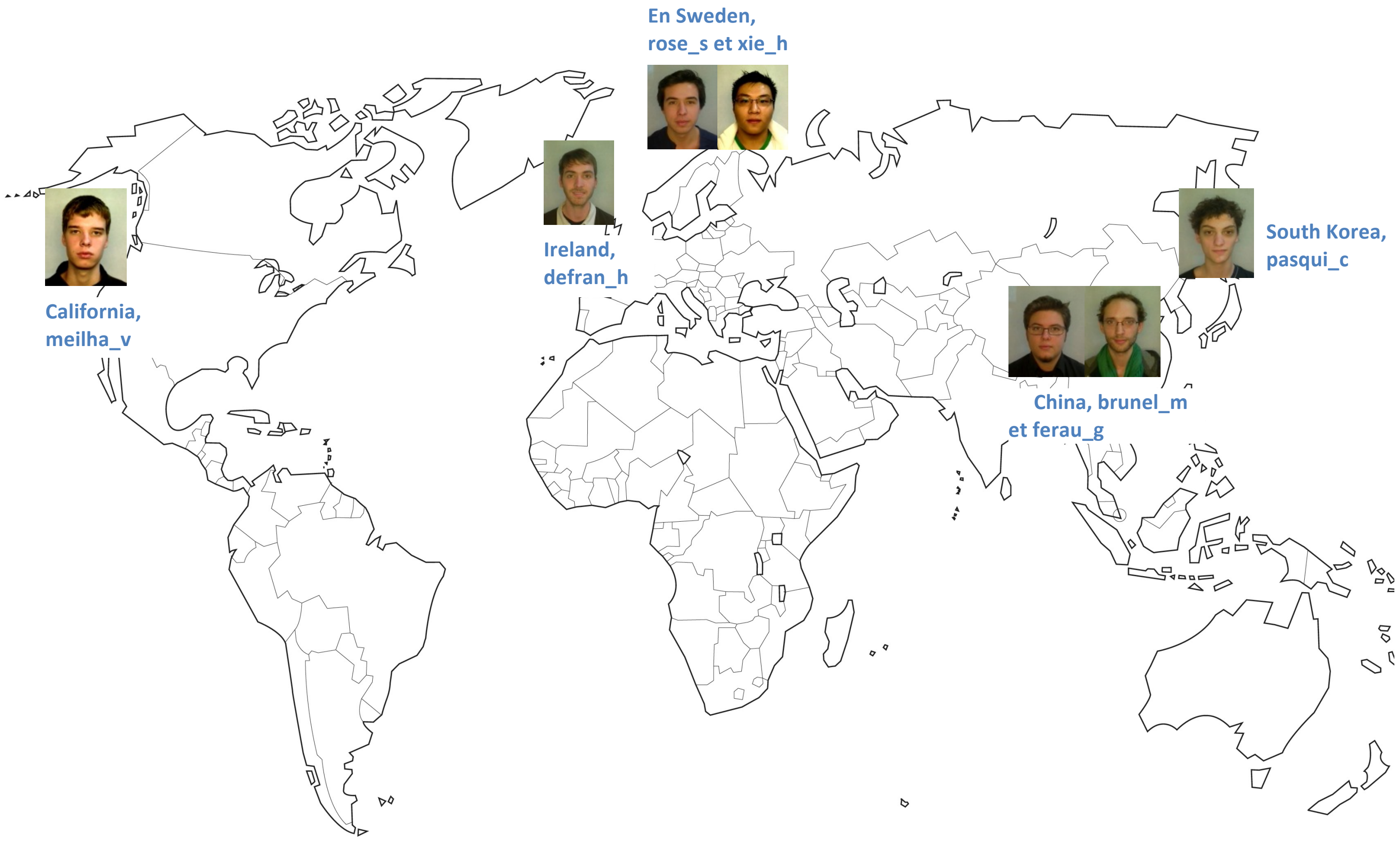
Title	AudioWire – Specifications
Date	18/04/2012
Authors	Huai Xie, Stéphane Rose, Guillaume Feraud, Maximilien Brunelle, Augustin Pasquini, Hugo Defrance, Vivien Meilhac
E-Mail	audiowire_2014@labeip.epitech.eu
Subject	Specifications
Key words	
Version	1.1

Table of Revisions:

Date	Authors	Section(s)	Comment
04/04/12	All group members	All	First Specification version
18/04/12	All group members	All	Second Specification version
04/07/12	All group members	All	Third Specification version

Table of Contents

Where are we?


En Sweden,
rose_s et xie_h
Ireland,
defran_h
California,
meilha_v
South Korea,
pasqui_c
China, brunel_m
et ferau_g

1. About the EIP

a. Epitech, school of passion and innovation

Epitech is an Information Technology school created in 1999, following Epita’s spirit. It holds passionate students that wish to learn computing through practical exercises instead of theory. It’s a 5 years long formation, that supplies an IT expert degree, recognized as a level one diploma by the CNCP.

The strength of this school is based mainly on the fact that students are brought to learn by themselves the notions. Notions acquired with grouped mini-projects, and more ambitious ones like the PFA and finally the EIP.

At the contrary of the regular scholarship teachings, where they bring you up the knowledge, an Epitech student should look for it by himself, to solve ambiguous problems in which he wouldn’t have any notions. This method of learning gives a capacity of adaptability to any situation and allows a student to handle and adapt to teamwork.

b. Epitech Innovative Project

The Epitech Innovative Project, EIP, is a group-based project, quite long and innovative, in the computing domain. It has to be done on an eighteen months period, and be constituted by at least five students.

The EIP is a glimpse of the school’s pedagogy. It allows the students to seek and develop an innovating idea from scratch, with the goal to be finished in time, on both technical and administrative ways.

This project means a lot in the “adventure“ of an Epitech student, because it allows him to evolve from a student to a professional. Indeed, the success of the EIP will allow the main part of the students to find their way on the working market, to create their own company, or to join an existing project interested by the same idea.

c. Objectives

We must consider that the EIP is not only a project aimed to show all the knowledge a student learned at Epitech. Technology, especially the IT sector, evolves constantly. That’s why the knowledge itself isn’t that important, because it will soon become obsolete. One of Epitech’s main goals is to teach students to outrun technical aspects of a project and to focus on documentation and communication. This part is very important because non-Epitech personnel will be giving students grades based on their work during these eighteen months.

2. EIP’s subject

“AudioWire” is an evolved music player.

It will have the following features:

- Library Manager

- Playlists Manager

- "Smart" Playlists (uses a “Learning Algorithm” based on the tastes and desires of the user) with algorithms related to data mining.

- Synchronized playback on the local network (two computers in two places playing the same song at the same time - the song is not necessarily on both machines)

- Visualizer

- Ability to listen to radio stations and podcasts

This software will also offer more web-oriented functionalities :

- A chat and a buddy list

- File Sharing

- Sharing of tastes (playlists, smart playlists ...)

- Inter-user Streaming, that is to say the ability to listen to the same music at the same time between multiple users.

AudioWire will be compatible with Windows, Mac and Linux. We also plan to port it to Android and IOS.

3. Economical context and targeted population

Nowadays numerous audio players exist and offer a multitude of possibilities. All these players such as iTunes or Windows Media Player have similar functionalities (library managers, playlist managers, etc). To these are added additional features differencing them ("Genius" on iTunes, or certain features of Winamp, etc). It is in this context that Audiowire will grow, this is where it should show its superiority over other players.

Audiowire will be fully free, and will run on Windows, Mac, and Linux. Our goal is to compete with other known players by implementing the most successful features and adding new features (such as inter-user Streaming, synchronization on the LAN, etc.)

AudioWire is aimed for the general public. Indeed, we believe it is important to make our software as simple as possible. By developing our software keeping this fact in mind, we're making sure we can reach a larger portion of the population.

However, it is important that more advanced users can set other options, and we will leave a way to change these advanced settings, without messing up with an average user’s usage.

4. Program's sections description

The AudioWire project consists of different parts:

- Artificial intelligence:

Artificial intelligence is the main part of our EIP. That's why we have established a partnership with the ACSEL laboratory in addition to the one with the EIP's laboratory. The Artificial Intelligence will be able to classify data according to different factors that will be implemented:

o The mood of the user: selects it manually or program defined.

o Analysis of audio files, either audio streams, or metadatas.

o World news, and/or personal information are also taken into account : what time is it, what time of year is it, what has happened recently in the world, Am I alone, Is it night time, etc.

We will use the WEKA software to analyze the audio stream and to pull out the musical clues.

Recovery of audio streams is a substantial part of AI, and includes various data to be processed:

o BPM

o metadata

o the tempo

o Breaks.

o Simple Genre management

- **The client:**

AudioWire is a very technical project. However it is important to make a very friendly user interface. By doing this we will be able to "seduce" more users. The two members of the group who will take care of the UI will have to implement the artificial intelligence part that will be developed in parallel.

The development of the client will have to respect an action plan defined in UML diagrams and transitions. The client will provide access to a detailed user profile, and will use all the features of AI ergonomically.

- Mobile Applications:

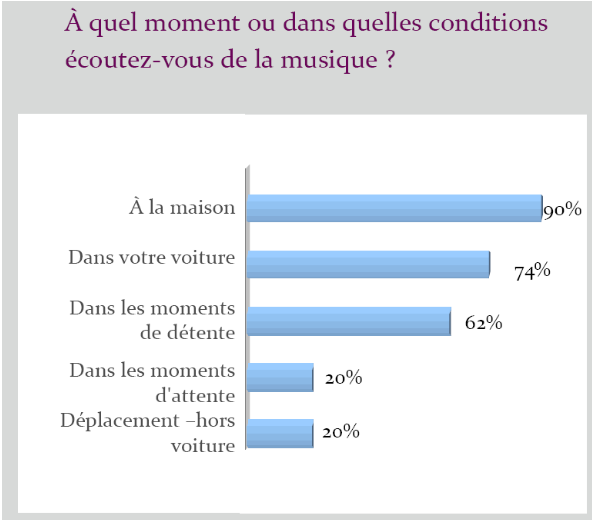
An important part of AudioWire will focus on mobile devices. We intend to develop an iPhone application and an Android application to play music intelligently. They will be available for free on their respective markets. These applications will integrate all the major features of the project, including music playback, depending on many parameters, user defined and artificial intelligence.

- Mobile Applications, why?

The development of these mobile applications will allow us to extend our project's usage.

Based on a survey by OpinionWay in 2011 (see annexes) taken on 2000 people, we find that music is the third cultural activity of the French population. Moreover, we see on the picture underneath that 62% of French people listen to music to relax, 20% during short travels (subways, buses…) and 20% of them listen to music while waiting. People aged from 15 years old to 24 admit they generally listen to an hour and a half of music a day, sometimes even more.

"When and where do you listen to music?"



That’s why we have decided to extend our software to mobile devices. Using this strategy, we will be able to gain more information on user’s tastes with a relation to their location. For example, we will be able to see that certain users prefer listening to jazz in the morning while on a subway ride and that they will enjoy rock music while going home.

A simple graphical interface will be developed for mobile devices as well, so that users won’t be lost with too many options.

- Database:

The project will lead us to store a very large amount of data, whether it be in music information and/or user data.

The EIP laboratory gave us the ability to borrow one of their servers to store all this information. This database will first be enriched by a list of songs provided by developers, then gradually by users.

5. Project details

a. Technical architecture

The media player will be usable on different platforms (Android, iPhone,Windows, Mac and Linux). These clients will communicate through a server that ensure liaison with the database.

A website is also planned to provide external communication around the project and allow users to use the social functions of the product. This website will communicate directly with the database. There will also be a development server that will be a true copy of theproduction server.

This will test all the new features before the start of production. This will guarantee us that our production server will always befunctional with a stable version of the project.

b. Existing components

To focus our efforts on artificial intelligence, the body of our media player will be based on the body of an existing project. In addition, we will use artificial intelligence algorithms to process audio data. These will be used to help our musical functions in our player.

c. Security questions

In order to preserve confidentiality, all transfers will be considered sensitive information and therefore be encrypted. TLS guarantees encryption of data transmitted through a system with private / public key.

Regarding the website, the non-secure connections (HTTP) will be automatically redirected in secure connections (HTTPS).

d. Sensitive Points

Because the nature of the project is largely based on artificial intelligence algorithms, it is conceivable that some of the media player options ultimately prove undoable. It is therefore important to evaluate the potential of existing algorithms to overcome the potential problems encountered as soon as possible.

How sensitive data will be transferred and stored will be subject to special attention in order to ensure confidentiality and security of them.

6. Utilities and constraints

a. Software utilities

To achieve our EIP, we will use the following tools:

- **Weka 3**: this is a software that we intend to use during the development phase of the project to achieve data-mining for audio files. This survey data will allow us to develop our artificial intelligence.
- **C++**: this language will be mainly used during development of our software on Windows, Mac and Linux.
- **Python**: Python is a scripting language that is both easy to handle and provides a functional and powerful code quickly. This language is mainly used for servers.
- **Java**: portable language with its own emulation environment. Java will be used to develop the Android client.
- **Cocoa/objective-C**: these two languages will be used for the GUI client for Mac and the entire IPhone client.
- **C#/.net**: this language and API will be used to carry out the graphics for the Windows client.
- **Virtualenv**: This is a tool to isolate a work environment in Python allowing developers to work on a common environment, regardless of the host machine. It will be used on the server managing access to the database.

To ensure good communication within the group, it will be necessary to use the following tools:

- Skype: For communication within the group.
- Microsoft Live Meeting: For communication between the laboratory and the EIP group.

b. Hardware utilities

In order to complete the project, we can distinguish two types of tools: hardware tools, and software tools.

- The hardware tools are represented by a server managing the database, the website and the various computers and telephones necessary for developers to work.

- The software tools necessary for developers to achieve the different clients of our project are IDEs (Integrated development environment). These will be Eclipse for the development of Android client, Microsoft VisualStudio for developing the Windows client and xCode for developing on Mac and iPhone clients. For Linux development, developers will be free to use their favorite IDE.

c. Constraints

As with all projects, our group will face different types of problems:

- **Location Problems:** developers will be located on different continents, which will therefore involve problems with our schedules.
- **Evolving project:** given that artificial intelligence is, by its nature, always evolving, it is likely that the latter will still be perfectible after the end of the project.
- **Resources problems:** the project is portable to many platforms; some developers will be unable to work on some of them due to a problem of limited resources.

7. Database

The database is a key point that should not be overlooked in the success of our EIP because it is what will allow the management of all information that will be needed both by user and by artificial intelligence. Indeed, the data indicated by the client will be directly saved to be used by different algorithms of research. It goes without saying that greater the database is, the greater the results will be.

Since the primary purpose of our EIP is not managing database but the realization of sophisticated algorithms in search of musical tastes, we have to choose an editor that is relatively simple to use and that allows us to spend as little time as possible in its design. Furthermore it is necessary that the research and treatment of this data is optimized so that the execution of our algorithms is shorter.

Our choice would be to model a MySQL or Oracle DBMS because these are the two most common models on the market and they are both cross-platform. However, MySQL would be more interesting given the nature of our project. It is true that Oracle is a powerful and highly efficient model for processing large databases but it is expensive and would be too big an investment especially in the context of our EIP.

That is why we think we will be using a Mysql database, as this choice would be more relevant in the context of AudioWire. Indeed, thanks to its ease of implementation and its effectiveness, this strategy will allow us to take advantage of a powerful and easy to use relational data model.

8. First level testing

AudioWire consists of two main parts: the first is the player itself, the second is the artificial intelligence algorithm that will allow an estimation of the user's tastes.

The first tests will take place on the player part because this part will have to be functional before others since it is the base of the project.

We will then test the following features:

- Library Manager

- Playlists Manager

- Visualizer

For these features, simple test will suffice. We will have to check their stability but also monitor their resources consumption (memory leaks, CPU usage, ...).

We will need, for these network function, to add to the tests described before a new list of tests to verify stability.

The implementation of scripts will be necessary and will allow us to optimize our functions and bandwidth consumption.

To allow synchronized playback we will have to do many tests and we will probably adapt to the network communication protocol.

Finally, the second part of the tests will be related to the AI algorithm. Throughout the EIP, we will test the viability of our technical choices with the members of the ACSEL laboratory. As an AI algorithm may require a "training" phase, our tests will be run simultaneously through small programs / scripts.

9. Project organization

At first we will establish two documents that will help us develop our artificial intelligence in our software. Those two reviews will give us overviews on Data Mining and on audio file analysis.

This part, in collaboration with the ACSEL laboratory will last until members of the group start leaving to foreign countries for their year abroad. At this point we will start developing the audio player.

At first we will develop the windows application. This strategy will allow us to develop our application and to distribute it to our friends at first. The major part of our project relies on artificial intelligence, and more precisely on Data Mining. This will allow us to establish what are the users' tastes, and it will allow us to offer them other songs.

Therefore it is essential that we recover as much information as we can, so that we can test our algorithm.

The windows platform imposes itself for the first release of our software because windows is still globally the most used operating system. By choosing it, we ensure that we will be able to have numerous users for our Beta testing period.

This Beta testing period will allow us to assemble a good database but will also allow us to find little bugs in program. This way, we'll be able to fix them before the final release of our software.

Once our Windows version is stable, we will start the second part of our development, which will consist of making the rest of the applications for the other platforms.

As for the windows platform, these applications will require testing periods.

10. Annexes

Sources concerning the databases (cf chapter 5):

Annexe 1:

http://fr.wikipedia.org/wiki/Base_de_données_relationnelle

The Wikipedia page about relational databases allow us to make not only a quick definition of the concept but also to better understand how it works and what it is made for in an IT project. We can then easily deduce the importance of a such structure and why it takes a such big place in the AudioWire project.

Annexe 2:

http://fr.wikipedia.org/wiki/Système_de_gestion_de_base_de_données

This link about the database management gives us an additional knowledge about the way a relational database communicates with the program itself.

Annexe 3:

http://webtic.free.fr/sql/

This document explains how a relational data model works and how to set it up. It especially explains the complexity of designing a database's structure and how important the database is.

Annexe 4:

http://www.commentcamarche.net/contents/sql/sqlintro.php3

This document explains the signification and the functionalities of the SQL language. It briefly defines how we should use it and when. Moreover, the SQL language is used a lot in the IT world, especially in the database world thanks to his request system that makes it easy to understand. Note that a direct concurrent exists: QBE (Query By Example). It is more instinctive to use (as a response matrix) but its way of writing request is heavier.

Annexe 5:

Weka: no, it's not the Chinese Wikipedia!

http://en.wikipedia.org/wiki/Weka_%28machine_learning%29

This Wikipedia page explains the origin of Weka: a pack of automatic learning softwares developed by the Waikato University in New Zealand. All the tools proposed by Weka allow us to do an analyze and a modeling of the data. The main advantages of Weka are that it's free, portable because developed in Java, and easy to use thanks to it's graphical interface.

Annexe 6:

www.sacem.fr/files/content/sites/fr/files/mediatheque/sacem/presse/etudes/etude_sacem/sondage_francais_musique/sondage_les_francais_et_la_musique_2011.pdf

Refer to the Gantt diagram.