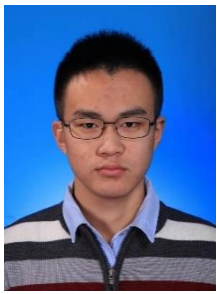




Music BRICK

Requirement Report

Group Assignment 4: Requirement Analysis

[1] Teammate

Photo	Name	Student ID	Responsibility (%)	email
	Shi Tianyang	1754191	33%	sty_lish@outlook.com
	Wang Jiaxuan	1754068	33%	1261494049@qq.com
	Gloria Dal Santo	1756133	33%	gloria.dalsanto@outlook.it

[2] Photo of Pair Team meeting

Team mates: I 4 panini



[3] Requirement report

[3.1] Introduction

[3.1.1] Background

Today, there is a lack of simple music editing websites for people with weak music foundations and basic knowledges. Furthermore, most of the software that are available nowadays are expansive, standalone (no web versions are present) and their free trial versions have too many limitations.

In short, there is almost no entry level software that can be used by amateurs.

[3.1.2] Purpose

Our project aims to solve the problem of the high cost of using music software and the limitations on the professional ability of common users. Basing on a webpage, we want to give the opportunity to whoever wants to edit music or create their own music and let them communicate with other users in order to improve their music skills.

[3.2] Requirement Gathering

[3.2.1] User Story Table with duration and priority estimation

Identifier	User Story	Size	Duration
ST-1	As a user who doesn't have music background, I want a simple software that can be easily used	4	8
ST-2	As a user who want to improve his skills, I want to experience some new and interesting elements (i.e. new effects, beats, ...)	6	12
ST-3	As a user who are not interested into the professional aspect of the creation of music, I want to have a simple introduction to it with a game	8	16
ST-4	As a user who wants to create his own music, I want to do it by using the product.	10	20
ST-5	As a user, I want to have the possibility to play in loop the music pattern.	8	16
ST-6	As a user who without any background in music, I want to learn some basic knowledges from the product.	4	8

0.5 points per day

Music BRICK

Requirement Report

[3.2.2] Functional Requirements Table with priority estimation

Identifier	Priority	Requirement
REQ-1	2	This webpage needs to have enough features to ensure that people without a musical background can use it. When someone logs in, he or she can follow the tips to achieve their willing.
REQ-2	3	The software needs to be designed to be interesting. The software should contain several entertainment functions (e.g. Puzzle Mode), which makes sure users have enough interest in it.
REQ-3	5	The software needs to allow users to replace the sound of instrument of a music snippet into another instrument. When given a music snippet, users choose which instrument they want to change, then the software can replace it according to user's request.
REQ-4	5	The product should give users some choices to ensure them to create their own music. When going to the edit page, users can choose the music factors they want to add and make them into a music snippet.
REQ-5	4	The product needs to have a function which can let a piece of music cycle in loop. When users give a music snippet and choose the function, the music can be cycled as long as they want.
REQ-6	4	The product should contain some part to make sure users can study some music knowledge in some ways. The page contains a communicate part to make sure users can share ideas with each other.

[3.2.2] Non-Functional Requirements Table

Identifier	Priority	Non-Functional Requirement
NFREQ-1	5	Strict permission access control, after the user has been authenticated, can only access the data within its jurisdiction, and can only perform operations within its jurisdiction.
NFREQ-2	5	Different users have different identities and permissions. They need to provide credible authorization management services under the premise of true and credible user identity, protect data from illegal/ultra vires access and tampering, and ensure the confidentiality and integrity of data.

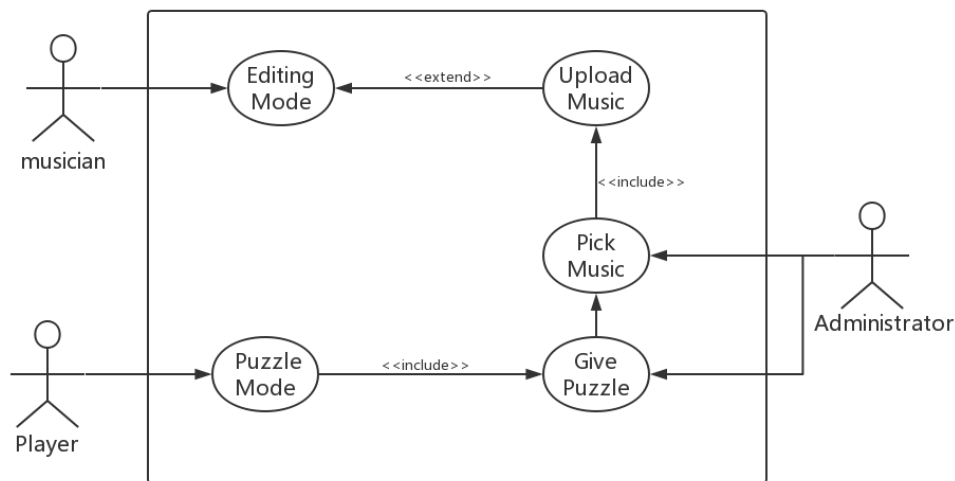
Music BRICK

Requirement Report

NFREQ-3	5	The network transfer data should be encrypted. It is necessary to ensure that data is not peeped, stolen, and tampered with during the collection, transmission, and processing. Business data needs to be encrypted when stored to ensure that it is not cracked.
NFREQ-4	4	The installation of the new version must keep all database contents and all personal settings unchanged. The product must provide a tool to track any database field.
NFREQ-5	3	There are hints for input, and data checks to prevent data anomalies.

[3.3] Requirement analysis

[3.3.1] Use Case Diagram of the system



[3.3.2] Example of Use Case Narrative

Use Case UC-1	Editing Mode
Related Requirement	REQ1 – REQ3 – REQ4 – REQ5
Initiating Actor	Musician
Actor's Goal	Create or Edit a song
Participating Actors	Administrator
Preconditions	The user must have done the login so that the system can load its account's information
Postconditions	The user can save its work. The system must require further modification and can upload its song on the software's database
Flow of events for main success scenario	

Music BRICK

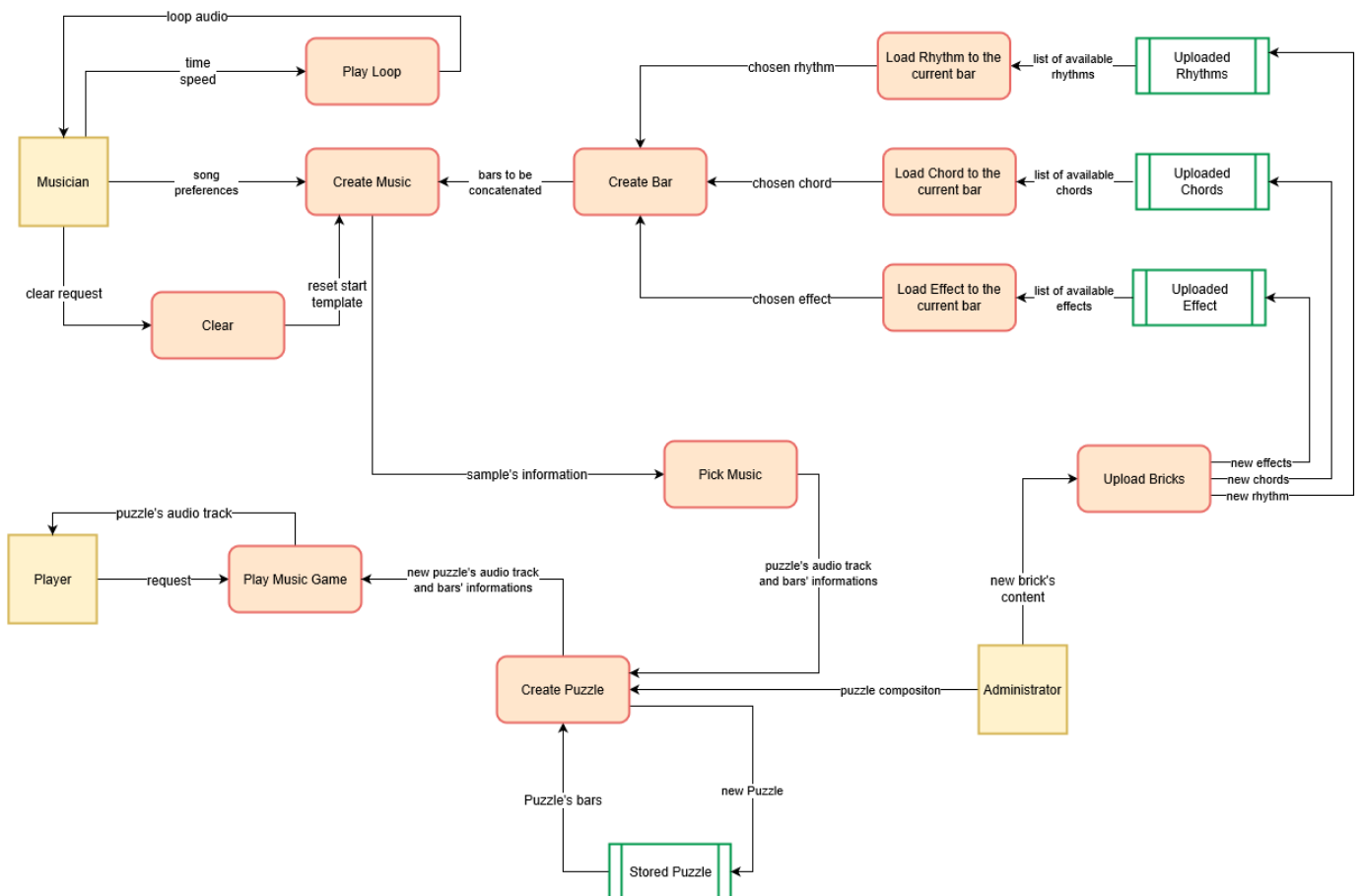
Requirement Report

->	Musician logged in successfully and request to have access to the Editing Mode
<-	System verify User's identity and load its information and provide all the Editing Mode's features
->	Administrator if required, provides new music to be selected by the user

[3.3.3] Use Case Priority Matrix

Identifier	Use-Case name	Priority
UC-1	Editing Mode	5
UC-2	Upload Music	3
UC-3	Pick Music	3
UC-4	Puzzle Mode	4
UC-5	Give Puzzle	5

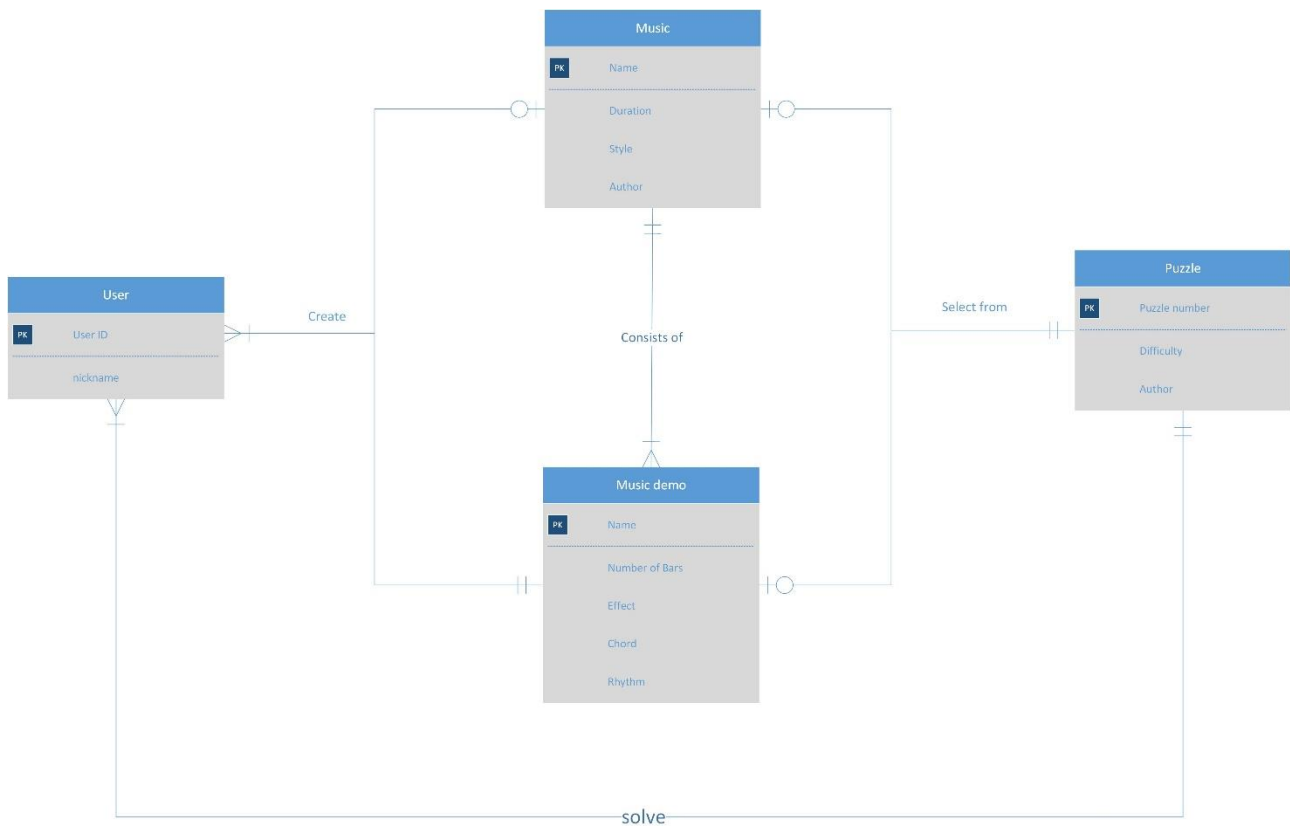
[3.3.4] Data Flow Diagram of the system



Music BRICK

Requirement Report

[3.3.5] Entity Relationship Diagram of the system



[3.4] On-Screen Appearance Requirement (UI)

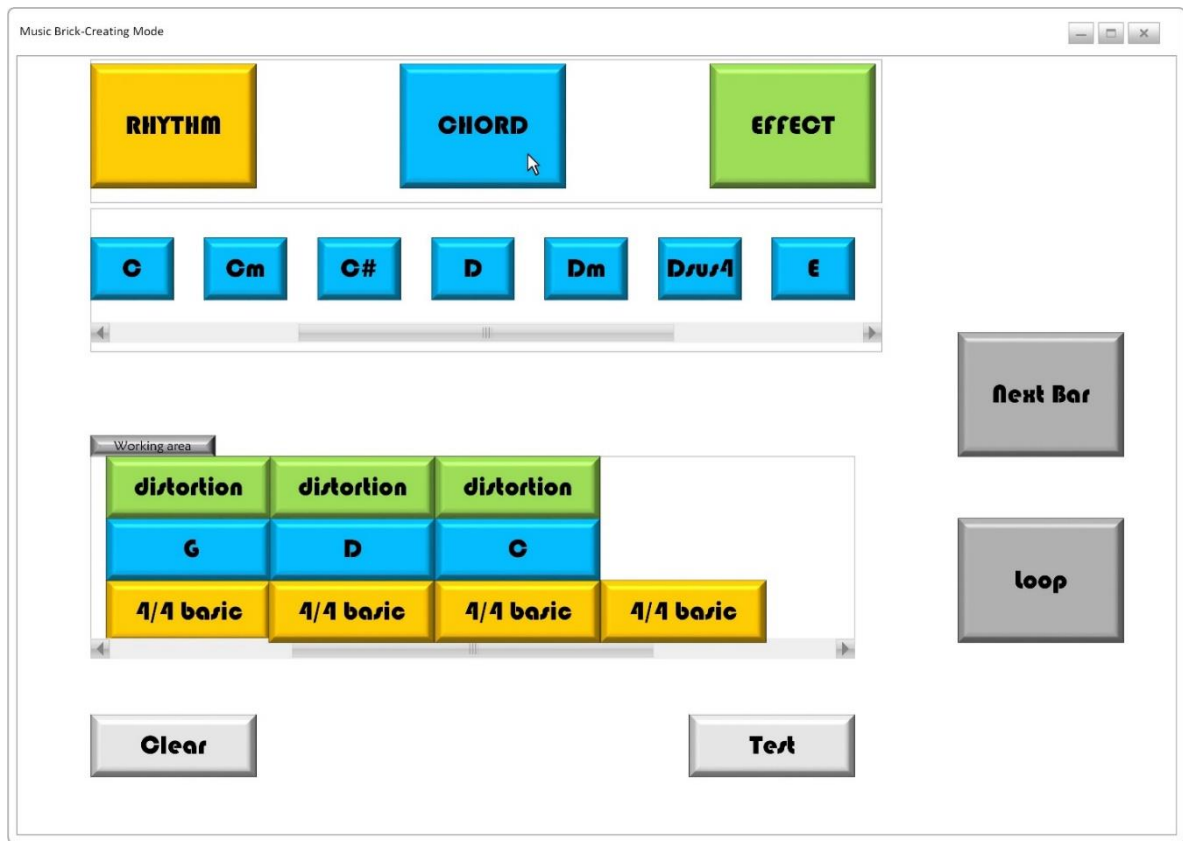
When users enter the program, they can choose to use creating mode (editing mode) or puzzle mode. The two options are showed as two colour bricks which can be clicked.

In creating mode, on the top of the screen are three bricks with different colour. They represent the rhythm, chord and effect of a single music unit. When clicking the bricks, there will be a column under them, which shows more bricks of more options. For example, when users click the brick of “chord”, the column will show dozens of chords like C, Am, Dsus4, etc. When users click one of them, it will play the sample sound of that chord. Double click the brick to select it for the music unit, and the colour bricks will be laid. It works in the similar way if users click the bricks of “rhythm” and “effect”. After selecting all the three elements for the music unit, users can click “Next Bar” to extend the music unit. After editing the music unit, users can click “Loop” button and select the basic information of looping like speed and time to repeat the music unit and make a whole piece of music. There will also be “Clear” (remove all the selection) and “Test” (play the sound of current selection).

In puzzle mode, the users can choose stage buy clicking the bricks displayed on the screen. Then the interface is almost the same with creating mode. The difference is that the buttons of “Loop” will be replaced by “Play” (which will play the puzzle sound by clicking it), and the choices of each element will be showed less. The users need to recover the puzzle sound.

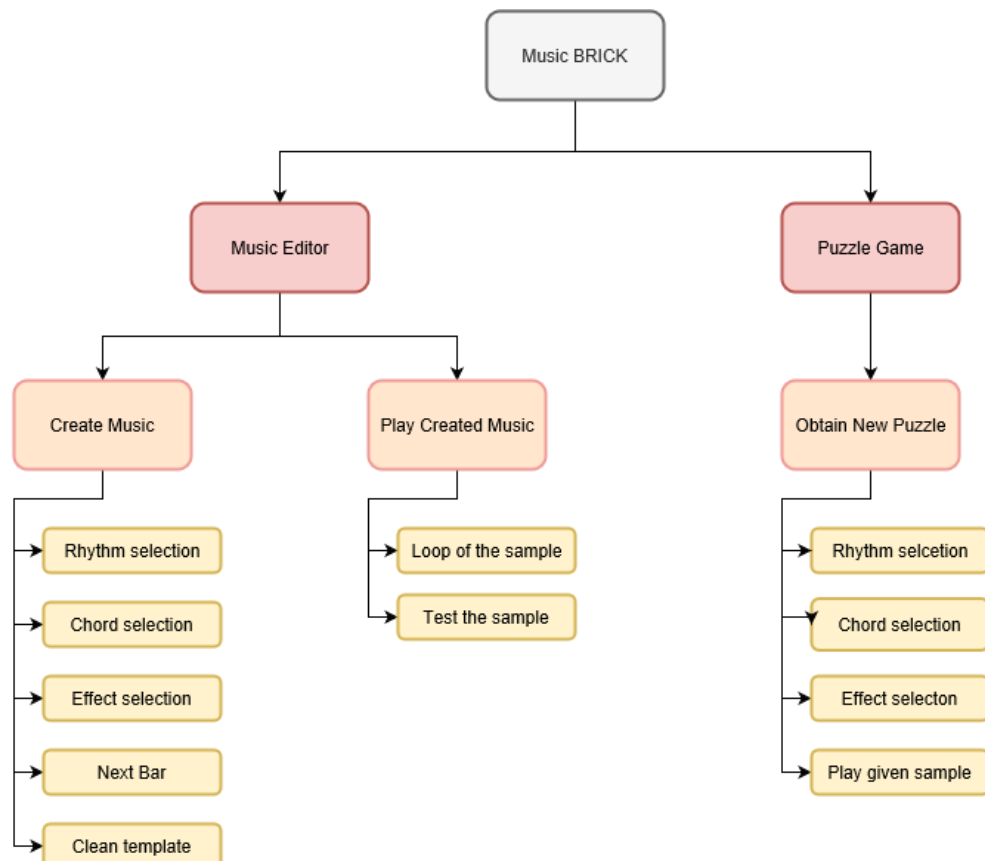
Music BRICK

Requirement Report



[3.5] Project Time Management

[3.5.1] Work Break Structure of the system

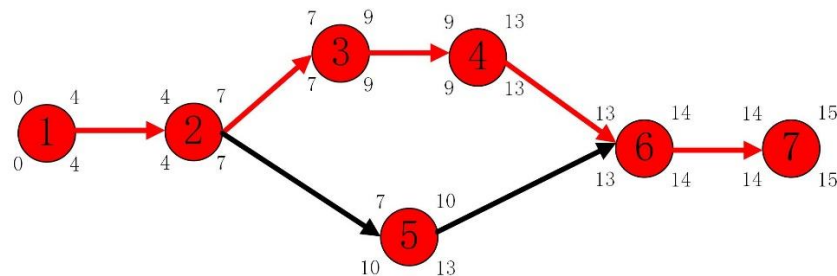


Music BRICK

Requirement Report

[3.5.2] Critical Path of the system

No.	Activity description	Predecessors	Estimated time(week)
1	Requirement analysing	0	4
2	Architect plans	1	3
3	Interface &framework designing	2	2
4	Basic function construction	3	4
5	Collecting materials	2	3
6	Preparing puzzle	4,5	1
7	Embellishing	6	1



The red arrows show the critical path.

[3.5.3] Slack Time of the system

No.	Activity description	ES	EF	LS	LF	Slack
1	Requirement analysing	0	4	0	4	0
2	Architect plans	4	7	4	7	0
3	Interface &framework designing	7	9	7	9	0
4	Basic function construction	9	13	9	13	0
5	Collecting materials	7	10	10	13	3
6	Preparing puzzle	13	14	13	14	0
7	Embellishing	14	15	14	15	0

(all measured by weeks)

[3.6] Conclusions

We think in this way we can build a functional software with affordable complexity with respect to our background knowledge.

The next steps require to analyse the program structure and practically how to develop it using Java Scrip language starting from the front-end interface construction.