# Team X

Requirements Document

Rebecca Young, Lineker Tomazeli, Alicia Bendz

November 4, 2011

COMP361 Software Engineering Project

School of Computer Science

McGill University

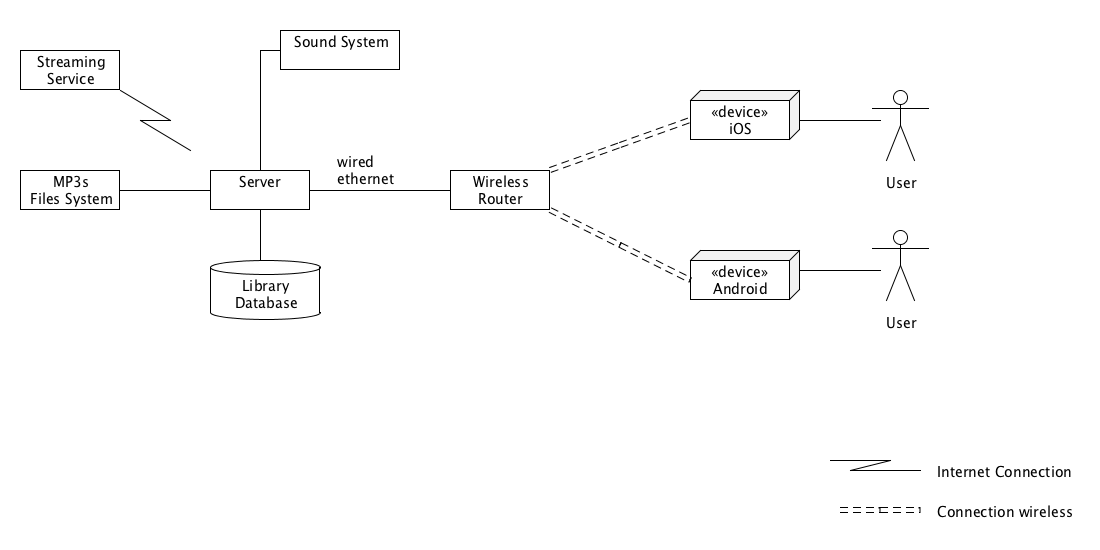
**SUB TITLE**

Subcategory

I. Introduction

II. Overall Description

Environment diagram



The environment diagram above gives a representation of the physical environment, and the software application required for our application to be functional.

The minimum requirement for our application is to have a computer to act as a server. The server has to be connected to the background sound system (speakers) and have at least some music files (mp3) in the local computer or a internet connection so streaming can be activated.

By having access to the music files or streaming, we can start using the server to manage libraries and play music.

By importing songs and managing libraries the database will be populated, creating relationships between songs and different libraries that the administrator can create. This data will also be available for statistical purposes.

To extend the application for user interaction with the sound system, it is required to have a router wireless where users can connect their devices to the network and communicate with the Server. The server must be connected to the same router and network that the users' devices will be connected. Devices running iOS or Android OS will be supported. The user has to download an application which will be available in the Apple Store and Android Market Place.

The administrator of the server can make libraries available to users, so they can suggest songs to the current playing queue.

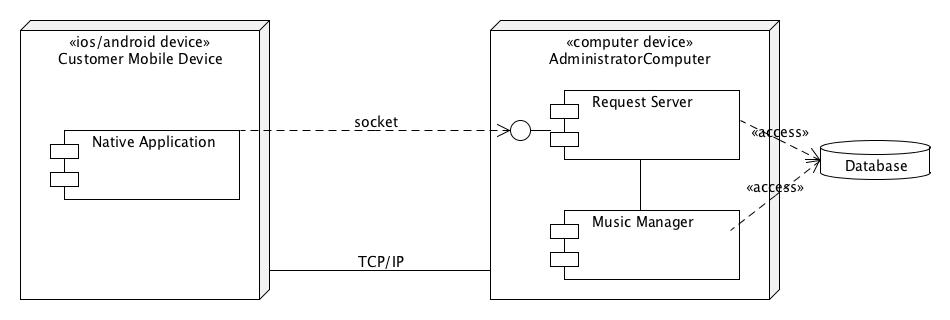
The user will also be able to send a feedback message to the administrator, increasing the interaction between users and administrator.

The Server can fully work without the internet, but the streaming feature will be deactivated, which forces the administrator to have their music files available on the file system.

At the moment, the only way of communication between clients and server is through the iOS and Android OS. No support is available to other platforms at the moment.

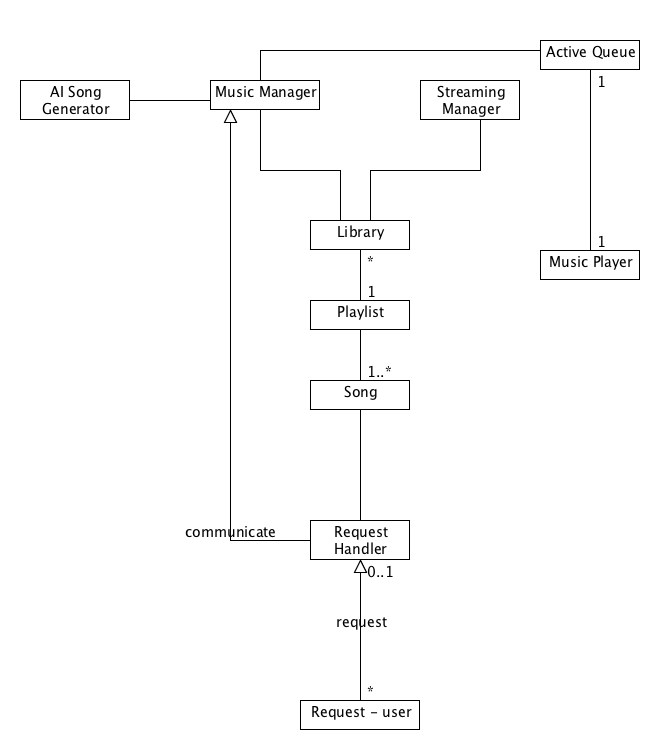
We deny any responsibility of setting up the local network and internet connection.

Installation diagram



Missing text…

Domain diagram



Alicia’s text…

III. Specific Requirements

Describe purpose of this section to the reader and how they should be reading it.

**3.1 Global Program View**

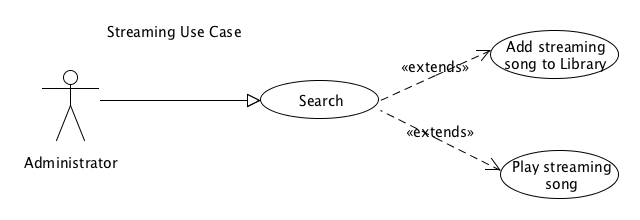
Menu diagram

Lecture 7.2

Global data-flow diagram

**3.2 Logical Unified Software Breakdown**

Module X: Streaming Manager



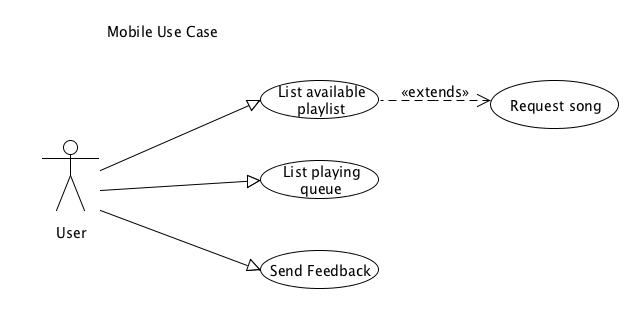
**Use Case Templates**

|  |  |
| --- | --- |
| Use Case Name: | Search |
| Iteration: |  |
| Summary: | The administrator wants to add songs from the streaming service to the library so first he has to search for it |
| Basic Course of Events: | 1. This use case begins when the administrator click on the search textbox 2. type name of artist, song or album that he is looking for and click search 3. the system respond with a list of songs found 4. See use case “Add Streaming song to Library” |
| Alternative Paths: | 4. Administrator can decide to play the song instead of adding to library. See use case “Play streaming song “ |
| Exception Paths: | If the system doesn’t find any songs it will return an empty list. |
| Extension Paths: | If the internet connection is interrupted it will not be possible to search for songs. I message will alert the administrator. |
| Triggers: | The administrator has decided to search for a song in the streaming service |
| Assumptions: | The computer running the application has access to the internet |
| Preconditions: | The administrator search for an artist, album or song name that exist. |
| Post conditions: | The administrator will receive a list of items found. |
| Related business rules: | none |
| Author: | Lineker Tomazeli |
| Date: | October 25, 2011 |

|  |  |
| --- | --- |
| Use Case Name: | Add Streaming song to Playlist |
| Iteration: |  |
| Summary: | The administrator wants to add songs from the streaming service |
| Basic Course of Events: | 1. This use case begins with the administrator searching for a song (see use case “Search”) 2. System respond with a list of songs found 3. Administrator select a song and clicks on “add” button 4. System will add song to current playlist and create references in the database 5. Song will be showed in the current playlist 6. Administrator can repeat steps 1-5 again if needed. |
| Alternative Paths: | 2. Administrator can double click the song he wants, this action replace the functionally of clicking in the “add” button |
| Exception Paths: | If the internet connection is interrupted it will not be possible to add the song. I message will alert the administrator. |
| Extension Paths: | none |
| Triggers: | The administrator has decided to add a streaming song to his library by selecting the song from the list found and clicking in “add” or double clicking the song. |
| Assumptions: | A search was made and records were found. Please see use case “Search” for details. |
| Preconditions: | The administrator search for an artist, album or song name and select one of the records found |
| Post conditions: | Song is added to the current library. |
| Related business rules: | none |
| Author: | Lineker Tomazeli |
| Date: | October 25, 2011 |

|  |  |
| --- | --- |
| Use Case Name: | Play Streaming song |
| Iteration: |  |
| Summary: | The administrator wants to play a song from the streaming service |
| Basic Course of Events: | 1. This use case begins with the administrator searching for a song (see use case “Search”) 2. System respond with a list of songs found 3. Administrator select a song and clicks on “play” button 4. System will start streaming (playing) the song, if there was a song playing before it will stop 5. Song will be showing in the current playing queue |
| Alternative Paths: | none |
| Exception Paths: | If the internet connection is interrupted it will not be possible to play the song. I message will alert the administrator. |
| Extension Paths: | none |
| Triggers: | The administrator has decided to play a streaming song by selecting the song from the list found and clicking in “play” |
| Assumptions: | A search was made and records were found. Please see use case “Search” for details. |
| Preconditions: | The administrator search for an artist, album or song name and select one of the records found |
| Post conditions: | Song starts to play and is added to the current playing queue. |
| Related business rules: | none |
| Author: | Lineker Tomazeli |
| Date: | October 25, 2011 |

Module X: Mobile Application



**Use Case Templates**

|  |  |
| --- | --- |
| Use Case Name: | List available playlist |
| Iteration: |  |
| Summary: | The user wants to view what songs are available to request. |
| Basic Course of Events: | 1. This use case begins with the user opening the mobile application (iOS or Android) 2. Select tab “Playlist” 3. The system communicates with the server asynchronously. 4. The system responds with a list of available songs. |
| Alternative Paths: | none |
| Exception Paths: | If the system doesn’t find the server it will return a empty list and alert the user |
| Extension Paths: | User can also “pull down” the list to refresh. |
| Triggers: | The user has decided to view the list of songs available for request. |
| Assumptions: | There is a server running in the network. |
| Preconditions: | The user has an device with the application installed |
| Post conditions: | The user will receive a list of songs available. |
| Related business rules: | none |
| Author: | Lineker Tomazeli |
| Date: | October 26, 2011 |

|  |  |
| --- | --- |
| Use Case Name: | List playing queue |
| Iteration: |  |
| Summary: | The user wants to view what songs will be played. |
| Basic Course of Events: | 1. This use case begins with the user opening the mobile application (iOS or Android) 2. Select tab “Playing Queue” 3. The system communicates with the server asynchronously. 4. The system responds with a list of songs. 5. The system will highlight the current playing song and show the 5 last played and the 3 next songs |
| Alternative Paths: | none |
| Exception Paths: | If the device is connected to a network but the system doesn’t find the server, it will return a empty list and alert the user  The device is not connected to any network, it will alert the user. |
| Extension Paths: | User can also “pull down” the list to refresh. |
| Triggers: | The user has decided to view a list of songs that were played, the song that will be played and the songs that will be played. |
| Assumptions: | There is a server running in the network. |
| Preconditions: | The user has an device with the application installed |
| Post conditions: | The user will receive a list of songs. |

|  |  |
| --- | --- |
| Use Case Name: | Send Feedback |
| Iteration: |  |
| Summary: | The user wants to send a feedback to the administrator |
| Basic Course of Events: | 1. This use case begins with the user opening the mobile application (iOS or Android) 2. Select tab “Feedback” 3. User types a feedback message and click “send” 4. The system sends the feedback message to the server asynchronously. 5. The system responds with a confirmation message saying that message was delivered. |
| Alternative Paths: | none |
| Exception Paths: | If the device is connected to a network but the system doesn’t find the server, it will return alert message to the user  The device is not connected to any network, it will alert the user. |
| Extension Paths: | none |
| Triggers: | The user has decided to send a feedback message to the administrator |
| Assumptions: | There is a server running in the network. |
| Preconditions: | User types a feedback message and click send |
| Post conditions: | Server save the message in the database and send a confirmation message to the user. |

|  |  |
| --- | --- |
| Use Case Name: | Request Song |
| Iteration: |  |
| Summary: | After user had listed the available playlist, he decided to request a song to play in the sound system. |
| Basic Course of Events: | 1. This use case begins with the user opening the mobile application (iOS or Android) 2. Select tab “Playlist”. See use case “List Available Playlist” for details 3. Select a song. 4. The system will send the user request to the server and responds with a confirmation message saying that message was added to the playing queue. |
| Alternative Paths: | 3. After selecting a song, the user has the option of specifying what time he wants to play it. |
| Exception Paths: | If the device is connected to a network but the system doesn’t find the server, it will return alert message to the user  The device is not connected to any network, it will alert the user. |
| Extension Paths: | The server might return a message saying that the request could not be completed. **Please see page xxx for possible problems.** |
| Triggers: | The user has decided to request a song |
| Assumptions: | There is a server running in the network. |
| Preconditions: | User chooses a song from the current playlist |
| Post conditions: | Song is added to the playing queue. |

Entity diagram with data dictionaries and file structures

Performance requirements

Design constraints and risks

Software attributes

IV. Supporting Information