|  |
| --- |
| **Institute of Technology Blanchardstown** |
| StreaMe |
| Live Streaming Cross-Platform Broadcasting Client |
|  |
| **Benoit Brayer, Nans Plancher, Romaric Delaunoy** |
| **Nans Plancher**  **Romaric Delaunoy** |
| **01/01/2013** |

|  |
| --- |
|  |

Submitted in part fulfilment for the degree of

**B.Sc. Computing**

School of Informatics and Engineering,

Institute of Technology Blanchardstown,

Dublin, Ireland

# Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Degree of **B.Sc. in Computer Science** in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

# Abstract

# Contents

[Abstract 1](#_Toc351030632)

[Contents 2](#_Toc351030633)

[Illustration Table 4](#_Toc351030634)

[I. Chapter 1: Introduction 5](#_Toc351030635)

[I.1 Introduction 5](#_Toc351030636)

[I.2 Project Design 5](#_Toc351030637)

[I.3 Objectives 5](#_Toc351030638)

[I.4 Motivation 5](#_Toc351030639)

[I.5 Technologies 5](#_Toc351030640)

[II. Chapter 2: Literature Review 6](#_Toc351030641)

[III. Chapter 3: System Analysis 7](#_Toc351030642)

[III.1 Overview 7](#_Toc351030643)

[III.1.1 Uses case 7](#_Toc351030644)

[III.2 Functional requirements 7](#_Toc351030645)

[III.2.1 Create a project 7](#_Toc351030646)

[III.2.2 Configure the streaming platform and parameters 7](#_Toc351030647)

[III.2.3 Add media sources into the project 7](#_Toc351030648)

[III.2.4 Send the stream to a server 7](#_Toc351030649)

[III.2.5 Display stream in the software 8](#_Toc351030650)

[III.2.6 Display feedback of the streaming 8](#_Toc351030651)

[III.3 Use Cases 8](#_Toc351030652)

[III.3.1 Create a new project 8](#_Toc351030653)

[III.3.2 Load the source available 8](#_Toc351030654)

[III.3.3 Add a source into the project 8](#_Toc351030655)

[III.3.4 Remove a source from the project 8](#_Toc351030656)

[III.3.5 Choose a broadcasting platform 8](#_Toc351030657)

[III.3.6 Configure the streaming parameters 8](#_Toc351030658)

[III.3.7 Save the project 8](#_Toc351030659)

[III.3.8 Open a project 8](#_Toc351030660)

[III.3.9 Start streaming 8](#_Toc351030661)

[III.3.10 Stop streaming 8](#_Toc351030662)

[III.3.11 Streaming display 8](#_Toc351030663)

[III.3.12 Streaming feedback 8](#_Toc351030664)

[IV. Chapter 4: System Design 9](#_Toc351030665)

[IV.1 User Interface Design 9](#_Toc351030666)

[IV.2 Functional Design 9](#_Toc351030667)

[IV.3 Classes Design 9](#_Toc351030668)

[IV.3.1 Design Pattern MVC 9](#_Toc351030669)

[IV.3.2 Overall Class Diagram 9](#_Toc351030670)

[IV.3.3 Class Diagram 9](#_Toc351030671)

[V. Chapter 5: Implementation 10](#_Toc351030672)

[V.1 Prototype Implementation 10](#_Toc351030673)

[V.1.1 Video Capture 10](#_Toc351030674)

[V.1.2 Video Broadcast 10](#_Toc351030675)

[V.1.3 Video Display 10](#_Toc351030676)

[V.2 Linux Implementation 10](#_Toc351030677)

[V.3 Installer on linux and windows 10](#_Toc351030678)

[V.4 Others ??? 10](#_Toc351030679)

[VI. Chapter 6: Testing and Evaluation 11](#_Toc351030680)

[VII. Chapter 7: Conclusion and Further Work 12](#_Toc351030681)

[VII.1 Further Work 12](#_Toc351030682)

[VII.2 Conclusion 12](#_Toc351030683)

# Illustration Table

[Figure III.1‑1 Use Case Diagram 6](#_Toc351028251)

# Chapter 1: Introduction

## Introduction

Description of the project

## Project Design

Brief description of how the project is use

## Objectives

What the system should be able to do

## Motivation

## Technologies

Technologies use

# Chapter 2: Literature Review

# Chapter 3: System Analysis

## Overview

Main objectives and key functionalities

The system must be capable of capturing video from camera such as a webcam and audio from a microphone of a computer and send a video and audio stream to a broadcasting platform to play it in live.

In order to address these issues six key elements were defined:

1. Create a project

The application must be capable to assist the user for the creation of a new project. Also the application must give the possibility to save and load configurations to the user by using project files.

1. Configure parameters

The possibility of fully configure the audio and video quality, and the size of frames before sending the stream must be given to the user but an simplified configuration must be available for users based on the user internet broadband speed.

1. Add media sources into the project

Adding video and audio source must be easy and intuitive.

1. Send the stream to a server

Sending audio and video to the broadcasting platform must be automatically available when the platform parameters are filled in.

1. Display stream in the software

A player must be present into the interface to show the content which is currently streaming.

1. Display feedback of the streaming

Feedback on the streaming must be given to the user into the interface.

### Uses case

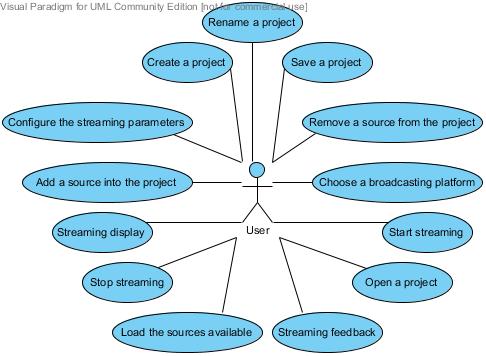


Figure III‑1 Use Case Diagram

## Functional requirements

### Store the project

1. Ensure the application can access the repertory where the user wants to save the project.
2. Ensure the project file is not corrupted when the application have to load it.

### Configure the streaming platform and parameters

1. Ensure to allow only coherent values using combo-box widgets and set a range of values for the upload broadband speed selection.
2. Ensure to limit the allowed platform to the platform where the application is able to stream to.

### Add media sources into the project

1. Ensure to not add two times the same source to the used source
2. Ensure that the is only one source for each type of source (video and audio)

### Send the stream to a server

1. Ensure that image and video codecs are the codecs allowed by the broadcasting platforms
2. Ensure to ask the user if something is wrong with the broadcasting platform like a bad streaming key, the broadcasting platform server down, no source defined as used.

### Display stream in the software

1. Use a Queue store each video and audio to play in the player and automatically reload this queue to keep playing the streaming since the streaming was not stopped

### Display feedback of the streaming

1. Ensure to display every useful feedback into the interface and to filter every useless feedback.

## Use Cases

For each use case: description, diagrams (main use case first)

### Create a new project

The user wants to create a new project.  
The user clicks on the “File” menu and click on the “create new project “button in the interface.  
A new window appears and asks the user to give a name to the project.  
After clicking on the create button, the new project is created and the streaming configuration windows appears.

### Load the source available

When starting a new project the sources available are automatically loads and appears in the sources list of the software

### Add a source into the project

When the user wants to add a source into his project he simply click on the source wanted in the source list and click on a button with an arrow to the right to add this source into the project, he can now see his selected source into the project source list call “Show”.

### Remove a source from the project

When the user wants to remove a source from the project he click on the source wanted into the source list of the project named “show” and click on the left arrow, he can see that the source is no more into the project list but in the source list.

### Choose a broadcasting platform

The user wants to choose a broadcasting platform. He clicks on the menu “Config”, then “Choose platform“, a window appear with the platforms he can choose, he can select a platform then enter a streaming key and click on the « ok » button. He can access to the streaming parameters too.

### Configure the streaming parameters

The user wants to configure the streaming parameters.  
The user clicks on the “Config” menu and click on the “Configure parameters” button in the interface.  
A new window appear witch allow the user to configure the broadcasting platform parameters (size, format, bitrate, speed). The user validates the changes by clicking on the ok button.

### Save the project

After configuring the project, the user can save it into a file, which can be reuse later. All parameters defined can be saved in the file like the sources selected the configuration the platform chosen.  
To save the project, the user has to go on the “file” menu, he can save if a project file is already create, or save as if a file doesn’t exist.

### Open a project

The User want to open an existing project, he go on the “file” menu and select “open a project “, he has to select the file of the existing project and click on ok.

### Rename a project

The user can rename a project when he want to, he just has to go in the “edit” menu and click “rename” a new window appears where he can enter the new name of the project and click on “ok” to validate it.

### Start streaming

The user wants to start streaming. If he has selected some sources for his project and configure the streaming parameters he can click on the play button in the main window or in the “Show” menu, then the streaming start and after a few seconds the stream is displayed into the main window.

### Stop streaming

The user can at every-time stop the streaming, by clicking on the button stop in the main window or in the “show” menu. Clicking on stop will stop the display of the stream and the broadcast on the streaming platform. When the user stops the streaming, the project is still available. If he wants to stream again he just has to click on the play button again.

### Streaming display

When the user wants to stream the result of what he sent is displayed into the software with just a little delay, the user can control the sound of the output.

### Streaming feedback

When the user is sending a stream he also has a text feedback into the software that will tell the state of the streaming and of the source capture.

# Chapter 4: System Design

## User Interface Design

Main window draft and description

## Functional Design

Description of the interactions ([Interaction overview diagram](http://en.wikipedia.org/wiki/Interaction_overview_diagram" \o "Interaction overview diagram) )

## Classes Design

### Design Pattern MVC

Description of the MVC and why we use it

### Overall Class Diagram

Description of classes

|  |  |
| --- | --- |
| Camera |  |
| Controller |  |
| Microphone |  |
| Project |  |
| Source |  |
| StreamThread |  |
| StreamTools |  |
| WinavTools |  |
| *ChooseCreateOpenProject* |  |
| *MainWindow* |  |
| *NewProjectAssistant* |  |
| *PlatformSelectionWindow* |  |
| *RenameProjectWindow* |  |
| *StreamingParametersConfigurationWindow* |  |

### Class Diagram

Class diagram

# Chapter 5: Implementation

## Prototype Implementation

One subpart for each main functionalities, explain of each is implemented

### Project File

For the storage of the project data we decided to use the XML file format to define our own file storage format policy. The goal of using this format is to make it readable also for humans using a simple text editor to simplify the debugging. Indeed, using this standard an experimented user can easily understand what configuration was selected by reading the project file. Also, the C++ framework we use (QT) include methods to create and read XML formatted files. These included methods accentuated our decision to use it because it also simplified the algorithm we had to perform to open a project from an existing project file.

Conform to the standard way of saving files into software, we decided to allow the user to save a project using two different way (save and save as). Also, to simplify the finding of the StreaMe project file, a StreaMe file type was defined (.sm). This file type helps the user when this one wants to open a project file because it allows filtering the files by this extension easily.

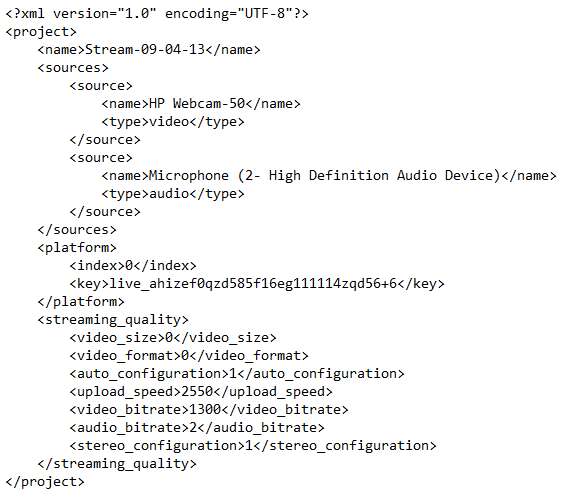


Figure 2 Example of project file

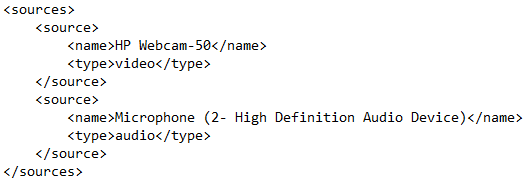
Into this storage file we choose to store each part of what defined the project itself.



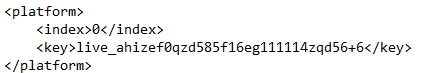
This line is just the xml header using for when loading the project file because this line define the encoding of the file and the xml version used. Without this line the Qt xml parser is not able to get the information form the file.



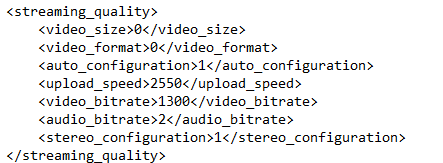
This line is just basicly the name of the project defined into the software. This name can be diferent that the name of the project file.



This part of the project file define each used source with his name as it apear into the software and the type of the source.



These information are the authentication information for the broadcasting platform.



Finally this part is the most important because it save every configuration for the audio and the video quality the user want to have when streaming the project.

### Video Capture

### Video Broadcast

### Video Display

## Linux Implementation

Description how the software is implemented on linux

## Installer on linux and windows

## Others ???

# Chapter 6: Testing and Evaluation

# Chapter 7: Conclusion and Further Work

## Further Work

## Conclusion