REAL TIME E-LEARNING PLATFORM

19-079

Software Requirements Specification

P.G.R.S.H. Gamlath

Dr. Malitha Wijesundara

B.Sc. Special (Honors) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

March 2019

REAL TIME E-LEARNING PLATFORM

19-079

Software Requirements Specification

P.G.R.S.H. Gamlath

Dr. Malitha Wijesundara

B.Sc. Special (Honors) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

March 2019

DECLARATION

I declare that this is my own work and this system requirement specification does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature	Date
P.G.R.S.H. Gamlath	IT16124936		

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of Supervisor Date

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide all the requirements and specifications for the ‘Video Player’ component of the ‘Real-time e-Learning Platform’. Main intention is to provide the information about requirements and specifications to the customers of the product. i.e. students, lecturers and management of the teaching institute. Apart from that, this document will provide the necessary details for the software engineers who are building and maintaining this product.

1.2 Scope

This document covers the requirements for and specifications of the Video Player component on initial release of the Real-time e-Learning Platform. But this document will contain some mentions about features that are expected to appear on future releases of the product. This will ensure the developers will choose appropriate design methodologies and strategies that will match for the whole project. So that this will facilitate a fluid flow of the development process without any abrupt interruptions along the path.

1.3 Definitions, Acronyms, and Abbreviations

UI	User Interface
GB	Gigabyte
MBps	Megabytes per second
RAM	Random Access Memory
HDD	Hard Disk Drive
ReLP	Real-time e-Learning Platform
ms	milliseconds

Table 1.3.1 – Acronym/Abbreviation

1.4 Overview

The Video Player is the main UI component of the Real-time e-Learning Platform which interfaces between the students and the system. It can be used to view live streams of the lecture sessions as well as playbacks of the earlier lecture sessions. The player will consist of multiple concurrent video feeds. Student can customize UI layout by maximizing any video of his/her choice while keeping the other video feeds minimized. A secure connection will be used for data transferring ensuring the copyrights of the videos.

Apart from the player functionality which are served for the students, there is an underlying functionality which will capture usage data in the background that will be processed in order to generate a detailed statistics report. This usage report will be used by the lecturers and management of the institute.

2 OVERALL DESCRIPTIONS

The Video Player component will be a web based player made on WebRTC protocols. This will ensure the usability of the application without much limitations because almost any device with a web browser and an active internet connection will be capable of running the application.

Player will consist of 2 video feeds namely, feed from the video camera specialized for tracking the lecturer and feed from the lecturers presenting screen which will contain teaching materials like slideshows. These 2 feeds will be in sync with the audio feed from the lecturer. There will be a 3rd video feed which will be activated only if a student is granted the permissions to ask a question from the lecturer, which in turn activates the student's webcam streaming hence providing the 3rd video feed. There is a separately developed whiteboard and a chat component which is integrated to the video player for the better usability.

All the views are always displayed except chat while one view of choice is kept maximized. Student can change the volume level and seek the video in a playback situation keeping the sync undisturbed. Lecture videos will not be able to downloaded as a complete video outside the player so that the rights of the videos are intact with the institute. User can change the video quality of the lectures as 360p, 720p or 1080p or set to 'Auto' so the player can decide which quality is the best based on device's internet speed capability.

In the background, player will keep a record of the video segments played and skipped as ranges on timeline. In a playback situation, it will also keep track of how many times each segment is played. Every time the maximized view changes, it records that change. So that, the lecturers and the management can have a good unbiased feedback about the lecture materials.

2.1 Product Perspective

A product with similar features to the Real-time e-Learning Platform called BigBlueButton is available in the current market. It contains some basic features common to Real-time e-Learning.

Feature	BigBlueButton	Proposed System
Real time streaming (WebRTC)	✓	✓
Multiple simultaneous streams	-	✓
Polling	✓	-
Dynamic Layout	-	✓
Usage Report	-	✓
Real time transcribing	-	✓
Closed caption (CC)	-	✓

✓ : Available

- : Not available

Table 2.1.1: Comparison with BigBlueButton

2.1.1 System Interfaces

- Web browser

2.1.2 User Interfaces

- Video Player interface
- Statistics report interface

2.1.3 Hardware Interfaces

- Mouse, Keyboard or Touch interface

2.1.4 Software Interfaces

- WebRTC API

2.1.5 Communication Interfaces

- Modem for internet access with speed capability of 3 MBps or higher.

2.1.6 Memory Constraints

- 1 GB or more RAM
- 2 GB or more free HDD space

2.1.7 Operations

- Play / Pause video
- Change volume
- Change video quality
- Choose maximized view
- Generate usage report

2.1.8 Site Adaptation Requirements

- Player interface must adapt to the display size of the device.

2.2 Product Functions

2.2.1 Use Case Diagram

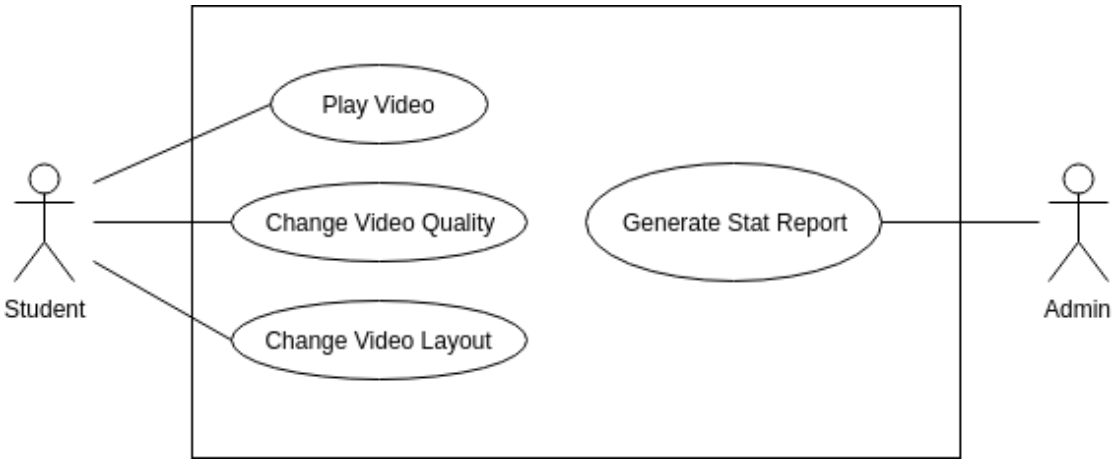

Figure 2.2.1.1: Use case diagram

2.2.2 Use case scenarios

Use case ID	UC1
Use case name	Play Video
Goal in context	Play the requested video (Live or Playback)
Primary Actors	Student
Pre-conditions	1. Student should login and arrive through a valid link. 2. Player should be fully loaded.
Main flow	1. Student clicks Play button. 2. System starts to play the video
Post-conditions	All the streams will start to play.

Table 2.2.2.1: Use case scenario 1

Use case ID	UC2
Use case name	Change Video Quality
Goal in context	Change the video quality of the main player
Primary Actors	Student
Pre-conditions	1. A video should be in play
Main flow	1. Student clicks the icon for the video quality. 2. System pops a list of available qualities. 3. Student select a quality. 4. System changes the source accordingly
Post-conditions	Main video player will change its video quality without interrupting the play.

Table 2.2.2.2: Use case scenario 2

Use case ID	UC3
Use case name	Change Video Layout
Goal in context	Change the video layout of the player
Primary Actors	Student
Pre-conditions	1. Player should be fully loaded.
Main flow	1. Student clicks on a sub player. 2. System interchanges clicked view feed with the main player view feed.
Post-conditions	All the feeds continue to play without interruption.

Table 2.2.2.3: Use case scenario 3

Use case ID	UC4
Use case name	Generate Stat Report
Goal in context	Generate a usage report based on data captured
Primary Actors	Admin
Pre-conditions	1. Sufficient data has to be available
Main flow	1. Admin clicks on a report type. 2. System displays set of filters available for report type. 3. Admin set the filters. 4. Admin clicks on generate report button. 5. System will display the requested report.
Post-conditions	-

Table 2.2.2.4: Use case scenario 4

2.3 User Characteristics

- Students and lecturers with the basic computer literacy.

2.4 Constraints

Video player shall function on any device with a web browser with WebRTC support ranging from mobile phone to computer regardless of the operating system given the device have the necessary processing power, memory and free storage space. The player shall be implemented in ReactJS framework.

2.5 Assumptions and Dependencies

- Users of the system shall have the basic skills of using a web based video player.
- Device should have the minimum hardware and software mentioned in 2.4 above.

2.6 Apportioning of Requirements

The requirements described in sections 1 and 2 of this document are referred to as primary specifications; those in section 3 are referred to as requirements (or functional) specifications. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both primary and functional specifications, the application will be built from functional specification since it is more detailed.

3 SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

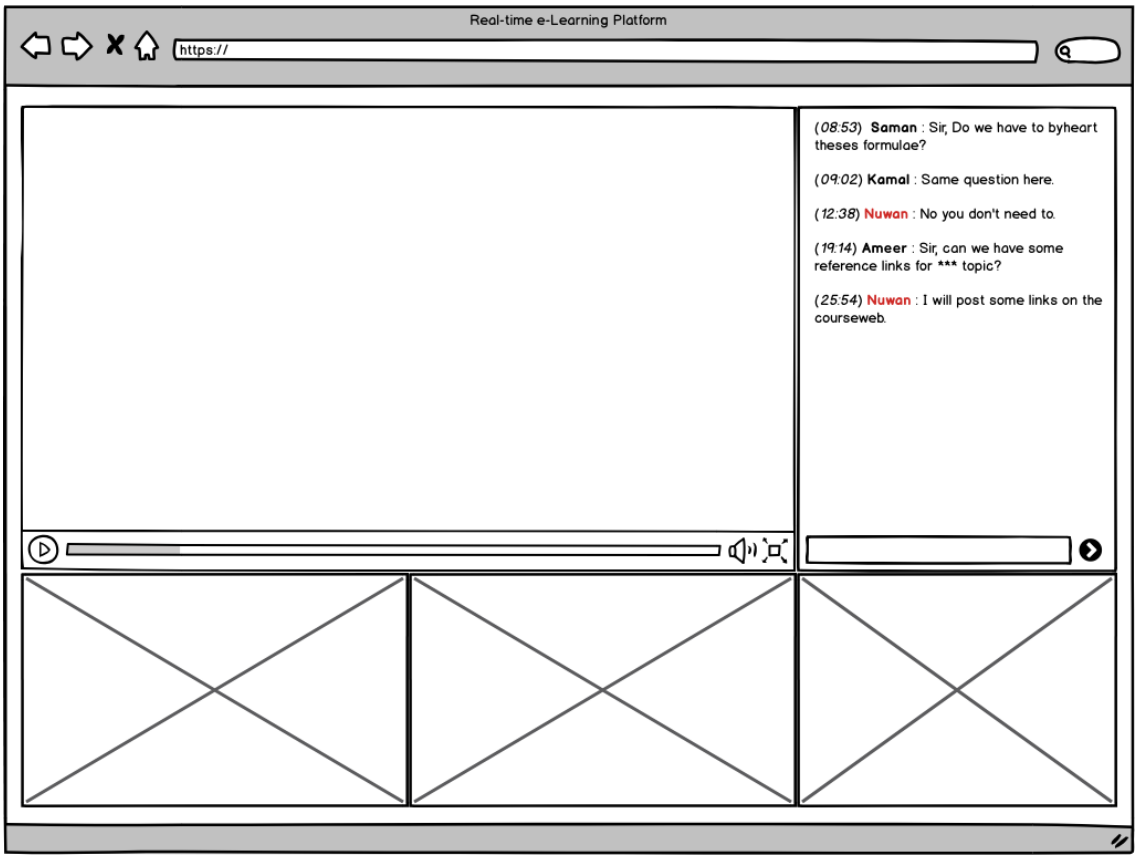

Figure 3.1.1.1: Video Player Interface

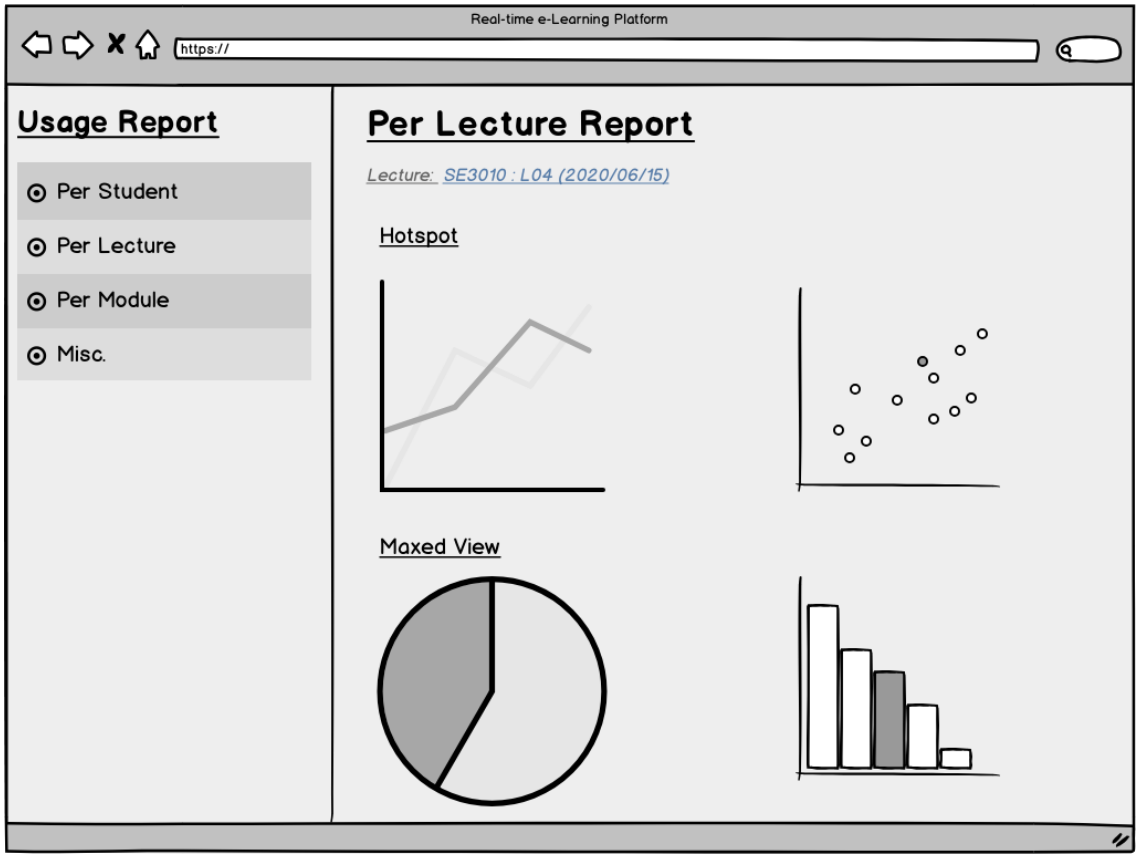

Figure 3.1.1.2: Statistics Report Interface

3.1.2 Hardware Interfaces

No special hardware is needed. If the device is a computer, a pointing device like mouse or touchpad is needed for the navigation purposes and a keyboard is needed for chats. No hardware devices are needed if the device has a touch screen.

3.1.3 Software Interfaces

WebRTC API will be used because of its wide support and its lack of special hardware or software requirement. Real-time video streaming and playing can be achieved inside a simple web browser.

3.1.4 Communication Interfaces

An internet connection will be needed as this is a web app. Due to heavy downloading of video streams and occasional video streaming requirements, a high bandwidth and low latency internet connection is required. This requirement will vary with the selected

quality of the downstream video. However, a connection speed above 3 MBps is preferred.

3.2 Classes/Objects

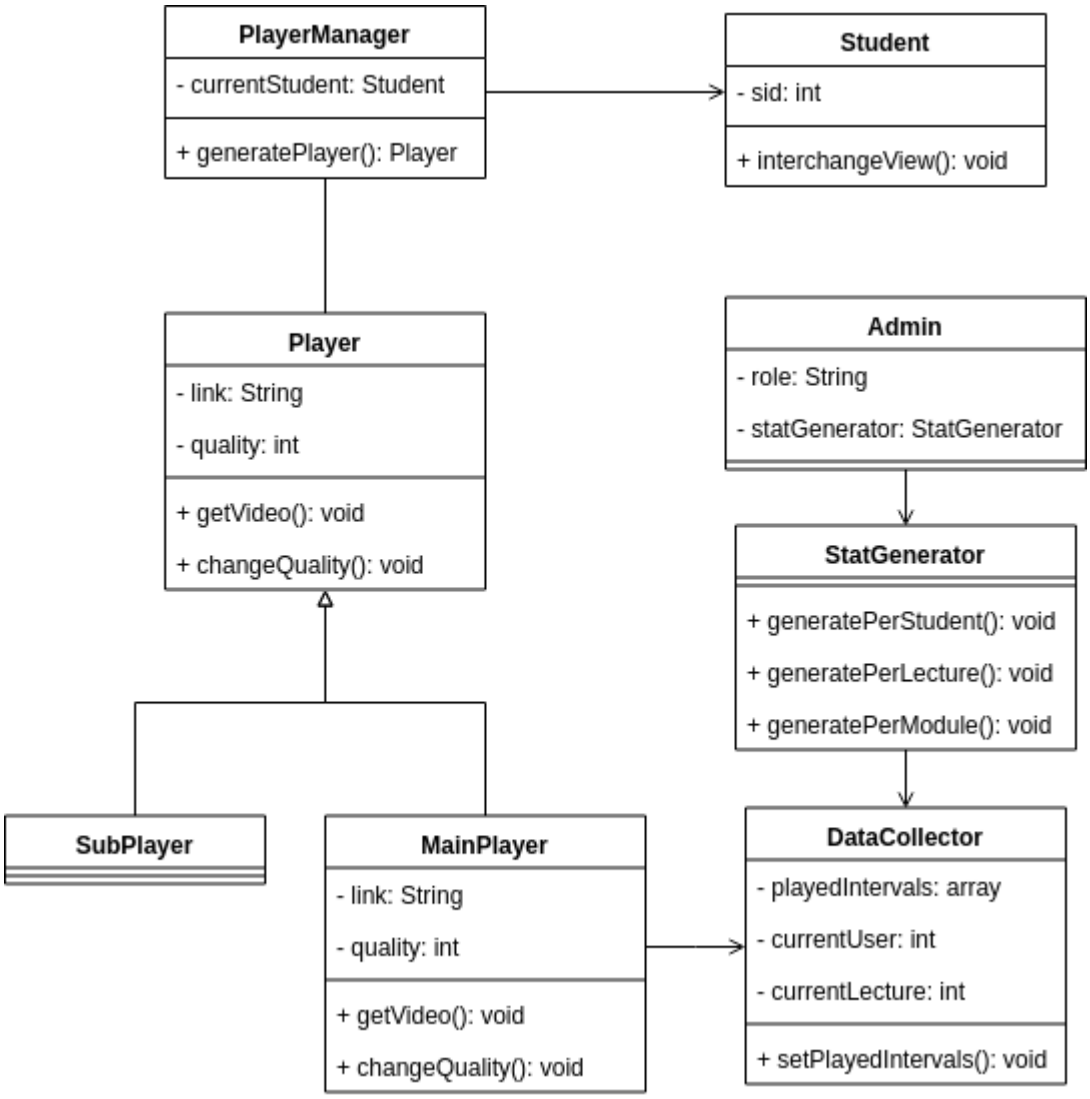

Figure 3.2.1: Class Diagram

3.3 Performance Requirements

Apart from the requirements specified elsewhere in this document, following requirements are expected.

- Product should perform without any lag under any stress. The highest stress is expected when the student stream from his/her device while viewing the down streams at full video quality.

- All the streams should be in sync at least at a level that is unnoticed by a human.
- Generation of any report should not take more than 1000ms.
- Whole system should load up and come to an idle state within 3000ms.

3.4 Design Constraints

A minimalistic design is required with a light theme.

3.5 Software System Attributes

3.5.1 Reliability

The system shall not crash unexpectedly under normal usage. Should display necessary information / warning / error messages appropriately.

3.5.2 Availability

System should check the availability of the server each 3 second and inform the user if availability is lost.

3.5.3 Security

The connection with the server should be secure rendering unauthorized downloading of video contents difficult.

3.5.4 Maintainability

The system should support maintainability. New report types should be able to add to the system without shutting down the whole system. Server address should be able to change without shutting the system down.

3.6 Other Requirements

As an optional feature there can be a transcriber module to generate a subtitle from the audio. This should happen at the time of play.

4 SUPPORTING INFORMATION

4.1 References

[1] Docs.bigbluebutton.org. (2018). BigBlueButton : Overview. [online] Available at: https://docs.bigbluebutton.org/overview/overview.html [Accessed 9 May 2019].