

Agora Streaming Platform: System Requirements Specification

Group 9:

Andrew Bennett 001319879

Joe Crozier 001311502

Nicholas Di Genova 400044942

Michael Panunto 400022970

November 4, 2018

Revision History

Revision	Date	Author(s)	Description
0.0	02/11/18	AB, JC, ND, MP	Creation
1.0	07/03/19	AB, JC, ND, MP	Applying feedback in accordance with TA feedback)
-	-	-	-

Contents

1. Introduction

- 1.1. Purpose
- 1.2. Scope
 - 1.2.1. Benefits
 - 1.2.2. Goals
- 1.3. Terminology
 - 1.3.1.
- 1.4. References
- 1.5. Overview

2. Diagrams

- 2.1. Context Diagram
- 2.2. Functional Decomposition Diagram

3. Functional Requirements

4. Non-Functional Requirements

- 4.1. Look and Feel Requirements
- 4.2. Usability Requirements
- 4.3. Operational and Environmental Requirements
- 4.4. Maintainability and Support Requirements
- 4.5. Security Requirements
- 4.6. Legal Requirements

5. Normal Operation

6. Undesired Event Handling

7. User Profiles and Personas

- 7.1. User Profiles
- 7.2. Personas

8. Future Developments

1. Introduction

1.1 Purpose

This system requirements specification document is intended to provide a layout of the features that shall be included in the final version of Agora. This document will include functional and non-functional requirements, typical operation of the website, and the website's target audience. This document is intended to be read by all stakeholders of the system.

1.2 Scope

The platform will be developed as a website compatible with all modern day browsers. It shall be designed with the user experience in mind, focusing on a simple Human Computer Interface/UX, ease of use, and information security.

1.2.1 Benefits

The benefits of Agora are:

1. Creation of an Information hub: Lectures on a variety of different topics will be hosted on a single platform, allowing users to learn about many different topics on one website.
2. Discoverability: Users will be able to easily find lectures/information on topics within/related to their field of interest.

1.2.2 Goals

The goals of Agora are:

1. Convenience
Agora aims to make attending and hosting lectures more convenient for the student by creating the experience of an in-person lecture online. With Agora, students and lecturers could attend/host lectures without leaving home, raising student attendance/reducing tardiness.
2. Educational Improvement
Agora aims to improve the lecture experience by giving students and lecturers access to tools and features not typically found in an in-person lecture.

1.3 Terminology

1.3.1 Lecture

A joinable room in which **lecturers** transmit audio, video, and text to **students** who are part of the lecture.

1.3.2 Class

A grouping of **lectures** hosted over a period of time.

1.3.3 Lecturer

The host of a **lecture**.

1.3.4 Student

A person who is attending a **lecture**.

1.3.5 Lecture Category

The category under which a lecture will fall. Examples may include mathematics, English, science, and history.

1.3.6 Public Lecture

A lecture which is discoverable and joinable by any student.

1.3.7 Public Class

A class which is discoverable and joinable by any student.

1.3.8 Private Lecture

A lecture which is only visible and joinable by students who have joined the class for which the lecture is a part of.

1.3.9 Private Class

A class which is only visible and joinable by students who have received an invitation by the lecturer.

2. Diagrams

2.1 Context Diagram



Figure 1: Context diagram showing the Education Streaming Platform's relationship with external entities

2.2 Functional Decomposition Diagram

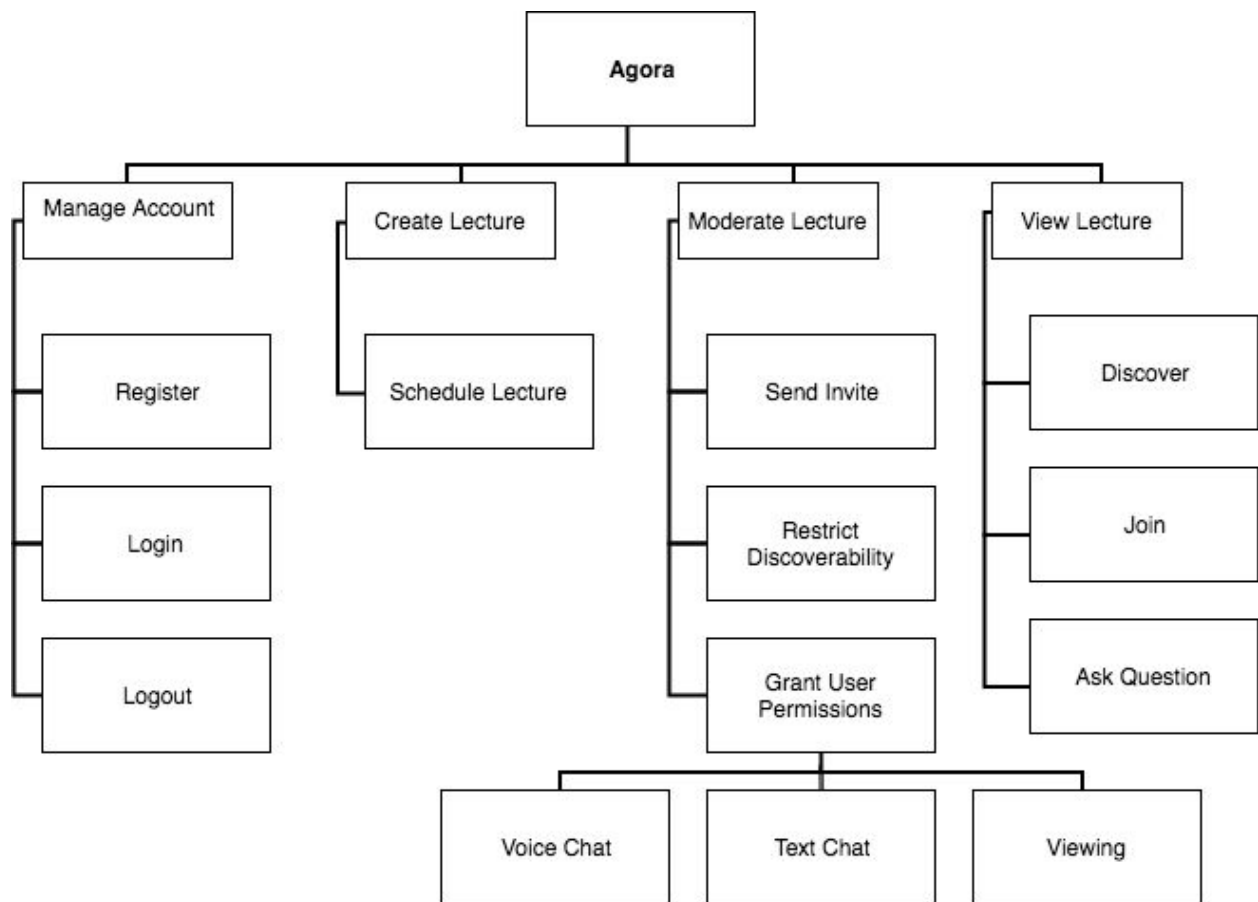


Figure 2: Functional Decomposition of Agora System

3. Functional Requirements

3.1 Lecture Facilitation Requirements

3.1.1 Lecture Hosting

A lecturer must be able to create a lecture to be joined by students.

3.1.2 Lecture Joining

3.1.2.1 Students can join lectures.

3.1.2.2 Students can only join currently running lectures

3.1.2.3 Lecturer shall be notified when a student joins their lecture

3.1.2.4 Lecturers shall be provided with a headcount in their current lecture

3.1.3 Connection

A websocket connection must be established in under 3.0 seconds after navigating to a lecture.

3.1.4 Data Transmission

Lecturers must be able to transmit audio, text, and video (hereafter referred to as data) to students presently in the lecture.

3.1.5 Student Request for Interaction

Students must have the ability to indicate to the lecturer that they wish to interact i.e., ask a question.

3.1.6 Interaction Request Acceptance/Denial

Lecturers must be able to accept or deny a student's request for interaction

3.1.7 Chat Room Moderation

A lecturer must be able to assign a moderator role to selected accounts.

3.1.8 Student Interaction

If allowed by the lecturer, the student must be able to transmit voice or text to the lecturer and all other students present in the lecture. The lecturer or the student must be able to end this transmission, allowing the lecture to return to its normal state.

3.1.9 Student Removal

Lecturers and moderators must be able to remove students from their lecture.

3.1.10 Chat Room Control

Lecturers and moderators must be able to mute/unmute students currently in the lecture.

3.1.11 Lecture Joining

Students must be able to join a lecture room at any time. If there is a currently ongoing broadcast the student will join it, otherwise a graphic containing the next lecture time will be shown.

3.1.12 Student Joining Notification

Lecturers must be notified when a student joins their lecture.

3.1.13 Lecture Headcount

Lecturers and students must be provided with a counter of the number of students currently in the lecture.

3.1.14 Lecture Attendance

Lecturers and students must be able to view a dropdown list containing the names of each student currently in the lecture.

3.1.15 Chat Room

There must be a chatroom that both students and lecturers can use to type messages publicly.

3.1.16 Chat Room Usernames

A lecturer's/student's username will be next the messages they send in the public chat room.

3.1.17 Chat Room Auto Scrolling

The chat room messages should auto scroll to the newest messages at the bottom of the message feed.

3.1.18 Lecture Length

A lecture must be able to run up to a maximum of 8 hours.

3.1.19 Lecturer Limit

A lecture must have only 1 lecturer hosting it.

3.1.20 Student Limit

A lecture must be able to support an indefinite amount of student viewers.

3.1.21 Host Limit

A lecturer must only be able to host 1 lecture at a time.

3.1.22 View Limit

A student must only be able to view 1 lecture at a time.

3.2 Lecture Discovery Requirements

3.2.1 Lecture Title

Lecturers must be able to add a title to their lectures.

3.2.2 Lecture Description

Lecturers must be able to add a description to their lectures.

3.2.3 Lecture Date/Time

Lecturers must be able to add a regular date and time for their lecture.

3.2.4 Lecture Category

Lecturers must be able to categorize their lecture.

3.2.5 Lecture Privacy

Lecturers must be able to specify whether they would like their lectures to be private or public. Private lectures cannot be discovered via search, and can only be joined if the student has been invited.

3.2.6 Class Invitations

Lecturers must be able to invite individual students using their username, or a group of students using a list of usernames separated by commas.

3.2.7 Lecture Browsing

Students must be able to browse through currently active lectures, filtering by category or title if desired.

3.2.8 Class Browsing

Students must be able to browse through classes which are currently being offered and will be hosting lectures in the future.

3.2.9 Lecture and Class Searching

Students must be able to use a text search to discover lectures based on their title, description, and category.

3.2.10 Search Result Filtering

Students must be able to filter search results by individual lectures or full classes.

3.2.11 Lecture Notifications

Students may enable lecture notifications for classes which they have joined. These notifications will be delivered via email, notifying the student of the start time for the lecture.

3.3 Authentication Requirements

3.3.1 User Sign-up

Users must be able to sign up for Agora as either a student or a lecturer. Users will be required to choose a unique username of at least 4 characters and a password of at least 8 characters.

3.3.2 User Login

Users must be able to login using their username and password to access features of Agora.

3.4 Interface Requirements

3.4.1 Discoverability

All actions which the user may take at any given instance must be immediately discoverable and appropriately labelled.

3.4.2 Interface Responsiveness

All actions which the user may take at any given instance must take place within 300ms of the user pressing the associated interactable.

3.4.3 Error Communication

Should an error take place, the user must be notified and informed of their correct course of action.

3.5 User Account Requirements

3.5.1 Name

Users must input their first and last names to be associated with their account

3.5.2 Date of Birth

Users must input their date of birth to be associated with their account

3.5.3 Email

Users must input their email address to be associated with their account

3.6 Robustness Requirements

3.6.1 Data Stream Error

Errors in the streaming of data that are not resolved in 3.0 seconds will cause the Agora client to attempt to open a new websocket connection and display a “Reconnecting” message to the user.

3.6.2 Stressful Environment

The Agora system will reject new video connections in times of unexpected load.

3.6.3 Process Recovery

The Agora system shall successfully reboot itself in the case of failure.

3.6.4 Data Recovery

The Agora system will perform database backups every 24 hours.

4. Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Simplicity

The site’s interface shall be simple and easy to navigate/find desired tools and functions.

4.1.2 Consistency

The styling of the user interface shall remain consistent throughout all pages on the site.

4.2 Usability Requirements

4.2.1 Ease of Use

The site shall be easy to use by the target users as defined in section 7 of this document.

4.2.3 Learnability

The site shall be easy to learn, requiring less than 3 minutes of browsing the site to enter an open lecture.

4.2.4 Understandability and Politeness

The site shall not contain any informal or curse words.

The site shall have the ability to revoke communication privileges from students who use inappropriate/rude language in the chat room.

The site shall have the ability to ban lecturers who use inappropriate/rude language in their lectures using the site.

4.2.6 Safety Critical Requirements

N/A

4.2.7 Precision Requirements

N/A

4.2.8 Availability Requirement

The system shall be available to access at all times, excluding maintenance downtime.

4.3 Operational and Environmental Requirements

4.3.1 Expected Software Environment

The site shall be compatible with the Google Chrome, Safari, Internet Explorer, and Mozilla Firefox internet browsers.

4.3.2 Input Devices

Users shall be able to use webcam and microphone to input video and audio data.

4.4 Maintainability and Support Requirements

4.4.1 Uptime

Agora must have an uptime of at least 99.9%.

4.4.2 Audit Trail

All changes made to the Agora codebase shall be recorded and timestamped using Git software.

4.4.3 Developer Commit Messages

All changes made to the Agora codebase through Git must be accompanied by an appropriate *commit* message that details clearly the changes made.

4.5 Safety and Security Requirements

4.5.1 Information Security

The account information of all users shall be encrypted.

4.5.2 Password Security

Account passwords of all users shall be hashed on the client before being sent to the Agora server.

4.5.3 Privacy

User information shall not be shared with any third party.

3.5.1 External Errors

Agora must not cause errors outside the constraints of its own runtime. This includes any errors which may cause instability in the browser or operating system.

3.5.2 Stability

Agora must not crash, become unresponsive, or enter into an undefined state.

4.6 Accessibility Requirements

4.6.1 Non-Text Content

Non-text content such as images and pre-recorded video that is presented to the user shall have a text alternative.

4.6.2 Text size

Except for captions and images of text, text shall be able to be resized without assistive technology up to 200 percent without loss of content or functionality.

4.6.3 Reading Level

Text and instructions on the Agora platform shall not require reading ability more advanced than a lower secondary education level to understand.

4.6.4 Error Display

If an system error is automatically detected, the item that is in error shall be identified and the error shall be described to the user in text.

4.7 Language and Internationalization Requirements

4.7.1 Locale

Agora shall support the language and customs of the *en_CA* locale.

4.7.2 Language

Agora's interfaces shall display text in the English language in accordance with English - CANADA (CA) (*en_CA*).

4.7.3 Date

Dates on the Agora platform shall be displayed in *y-MM-dd* or *EEEE, MMMM d, y* formats.

5. Normal Operation

Agora shall be primarily assessed on a desktop computer or laptop through one of the supported web browsers. There are no plans for mobile device support/optimization.

Agora shall only be accessible by users have registered up for the service, as defined in section 3.3.1.

6. Undesired Event Handling

During operation, errors may occur within the application. Krakowiak and Kaiser separated these errors into incidents and crashes, where incidents are undesired events that were successfully caught. Any other errors that a system is unsuccessful in catching are referred to as crashes [1].

In the case of Agora, gateway connection incidents, including loss of connection, session timeouts, and authentication failure shall be handled with an error message containing a description of the incident alongside a "Retry" button. Where as application crashes shall be handled by informing the user of the crash, while fetching the application logs in the background to allow for debugging.

7. User Profiles and Personas

7.1 User Profiles

Agora will cater to a variety of demographics with the common goals of learning or teaching others. A user of Agora can be a person of any age or gender that wants to:

- Attend lecture online as opposed to in-person
- Wants to teach lectures online to reach a larger audience.
- Wants to learn about random unfamiliar topics

7.2 Personas

7.2.1 Greg, University Professor

Greg is a 55 year old professor teaching at McMaster University. Greg teaches a second year calculus course at 8:30AM 3 times a week. He has noticed that less and less students are

attending his class, and midterm scores have been dropping. Greg wants the best for his students, and spends a lot of time thinking about a way to increase attendance.

7.2.2 Jake, Hardworking Student

Jake is a 20 year old Chemical Engineering student at McMaster University. To help pay for his education, Jake has gotten a part time job at a local restaurant, where he works late night shifts. Unfortunately, the late night shifts make it difficult for him to wake up early and commute to some of his early morning lectures, and as a result his grades have dropped significantly.

7.2.3 Simon, Curious Teenager

Simon is a 15 year old highschool student. Simon has always done very well in school and often finds himself very bored of his classes. He spends most of his free time searching the internet for informational or instructional websites on any topic at all, as he loves learning about new things.

References

[1] Kaiser, C., Krakowiak, S., "An Analysis of Some Rune-Time Errors in an Operating System", Colloques IRIA, Rocquencourt, April 23-25, 1974.

Feed back

~~-Put references at the end (DONE)~~

~~-It would be good to see a functional decomposition diagram with your requirements ab(DONE)~~

~~-Include likely to change requirements ab~~

~~-I think security relates more closely to safety than external errors and stability~~

~~-Think about how to filter out inappropriate things in chat, etc. maybe add a ranking system for comments.~~

~~-- Missing Maintainability Reqs Missing Security Requirements, Missing Robustness Requirements, Missing accessibility Requirements Missing Language and Internationalization Requirements. Likely changes not provided DONE~~

~~-References should be at the bottom of the document or the bottom of a page (DONE)~~

~~- floating headings, see section 4.5 REVIEW AT END~~

~~-context diagram too simple to be usefull missing things such as question responses~~

~~- Requirements have unnecessary details, eg: 3.1.6 the reuirement is actually just: "lecturers can remove students from lectures" the reason doesnt matter for the requirement other than as a rationale.~~

~~Insufficient requirements regarding the robustness of the data streams (not all video errors should produce a popup).~~

No requirements about the persistance of lecture information after the lecture has ended (are these lectures recorded for later viewing, are they only viewable once? Do they get marked as "watched" after they've been watched? Is information for every lecture ever done kept for eternity, or does the information "expire" after a certain time.

~~No requirements given about limiting access to private lectures/classes.~~

No requirements given for the visibility of which student is currently broadcasting.

No requirements for the type of network access that can be used, are these students remote? Are lectures viewable over the internet? Bluetooth?

~~What metadata can be set by the lecturers. (DONE)~~

What requirements are there on the storage of passwords.

~~how many lecturers are there per lecture? (DONE)~~

~~is there a limit on the number of students per lecture? (DONE)~~

~~Can there be a moderator person as well as a lecturer? (DONE)~~

~~How many lectures can a user participate in at the same time? (DONE)~~

~~how many lectures can a lecturer run at the same time? (DONE)~~

~~Is there a limit on the length of a lecture? (DONE)~~

~~Does a lecturer have to add students individually, or can they add them as a group? (DONE)~~

~~What video size can the system support?~~

~~What other data types can the system support? What are the use cases for them?~~

~~Does the system support recording via webcams? Microphones?~~

~~what are the network speed requirements for each participant?~~

~~Why can't students join a lecture before it begins? (DONE)~~

~~Are students automatically kicked out of the lecture when it ends? (DONE)~~

~~Can students see the names of other students in the lecture? (DONE)~~

~~Can students see the name of the lecture, class and lecturer, while in the lecture?~~

~~Does the Lecturer see this information while presenting?~~

~~Does the lecturer see the names of all the students while presenting? (DONE)~~

~~why is there no other data like name or email stored for users? (DONE)~~

~~Is there a chat function similar to Youtube Streaming? (DONE)~~

~~How many video feeds are supported?~~

~~Can lecturers switch between video feeds like at a normal podium?~~

~~No security requirements given regarding communications protocol security or access control.~~

~~Safety Requirements aren't actually safety requirements~~

~~-Requirements are too wordy and contain too many individual details, for eg: Req #3.1.2 is actually 4 different requirements (1: students can join lectures, 2: Students can only join currently running lectures, 3: Lecturer shall be notified when a student joins their lecture 4: Lecturers shall be provided with a headcount in their current lecture) (DONE)~~

~~-No rationale provided for many requirements: For example the 500ms latency number in 3.1.4 and 3.1.3, this number borders on being unrealistic in some scenarios when buffering is used~~