

---

# Software Requirements Specification

for

## LeTour Guide

Prepared by

**Group Name:** *LeTour Team*

*Richard Homan*

*Joel Wink*

*Cameron Bundy*

**Instructor:** *Prof. Doug Lim*

**Course:** *Software Engineering II*

**Version:** *1.0*

**Date:** *September 30, 2024*

**CLIENT SIGN-OFF**

---

*print name*

---

*signature*

---

*date*

<b>1 Introduction.....</b>	<b>1</b>
1.1 Document Purpose.....	1
1.2 Product Scope.....	1
1.3 Definitions, Acronyms and Abbreviations.....	1
1.4 References and Acknowledgments.....	1
<b>2 Overall Description.....</b>	<b>2</b>
2.1 Product Functionality.....	2
2.2 End Users and Characteristics.....	2
2.3 System Stakeholders.....	3
2.4 Operating Environment.....	3
<b>3 Specific Requirements.....</b>	<b>4</b>
3.1 User Interface.....	4
3.2 Functional Requirements.....	6
3.3 Use Case Diagram.....	8
<b>4 Non-functional Requirements.....</b>	<b>9</b>
4.1 Software Quality Attributes.....	9
<b>5 Other Requirements.....</b>	<b>11</b>

# 1 Introduction

LETU campus tours are an important experience for upcoming college students, parents, grandparents, and alumni alike. The campus tour is often the first experience interested parties have at LETU, so it is important that campus tours are comfortable for the audience. Campus tours are around 30 to 60 minutes long, and are led by a student. During preview events, held several times each year, the headcount on a single tour can vary from 20 to 50 people.

For large groups, especially in hallways where the tour may be compressed into a long line, it can be difficult for individuals in the back of a campus tour to hear the tour guide. It is important to LETU that everyone hears what the guide says. Mishearing the guide can be frustrating for the listener and can result in a more negative experience of the institution overall.

## 1.1 Document Purpose

The purpose of this document is to be a reference to create a simplistic solution to this unique problem. By addressing this problem, the audience can much better understand the guide, which generally improves the comfortability of the individual during the tour. This can lead to an overall positive first impression of LETU. Of course, a more comfortable experience will, in general, bring in more students, which improves the life of the stakeholders, but also brings in more peers to improve the life of the student.

## 1.2 Product Scope

A major constraint in making this solution is designing access to the solution as easy as possible. The solution is to be accessible by a web browser to eliminate the need for installing/uninstalling software. Although the product is accessible as a single website, it is split into two parts: the front end and back end. The front end shall be the server that provides the client web browser with user interface information and logic. The back end shall be the server that handles the audio processing.

## 1.3 Definitions, Acronyms and Abbreviations

UI: User Interface

GB: Gigabyte

RAM: Random Access Memory

## 1.4 References and Acknowledgments

- Janus WebRTC server demo: <https://janus.conf.meetecho.com/demos/audiobridge.html>
- ReactJS: <https://react.dev/>
- Docker: <https://www.docker.com/>

## 2 Overall Description

### 2.1 Product Functionality

- As long as the mobile user can run a web browser, the user can join a designated group and listen and engage with the tour guide
- Make the user experience engaging and not convoluted to use
- Needs to send continuous audio streams between tour guide and tourists
- Tourists (e.g. the back of a tourist group) can ask questions and the tour guide will hear them easier

### 2.2 End Users and Characteristics

#### Audience–Top Priority

- Use: every 3-4 months for up to 2 hours
- Max Quantity: 200 Users
- Audience Domain
- Minimal expertise required
- Requirement
  - Phone w/ web browser
  - Internet Connection

#### Guides–Top Priority

- Use: every 3-4 months for up to 2 hours
- Max Quantity: 10 Users
- Guide Domain
- Minimal expertise required
- Requirement
  - Phone w/ web browser
  - Internet Connection

### 2.3 System Stakeholders

- Mr. John Tixier – Project Lead
- LeTourneau Admissions Dept. – Managing Client
- LETU IT dept. – Maintenance Client

### 2.4 Operating Environment

The software will be run in a docker container for ease of use by the admin and within the docker container will have our frontend and backend server instances. The frontend is running a NodeJS

React server for the UI of our webpage and the backend is running a WebRTC-Janus server. The hardware requirements for the server are local network port privileges, being able to run a Docker container, at least 2GB+ storage space, and at least 2GB+ dedicated RAM. For the user device, a phone with a modern browser like modern versions of Edge, Chrome, and Firefox.

## 3 Specific Requirements

### 3.1 User Interface

#### Audience Domain

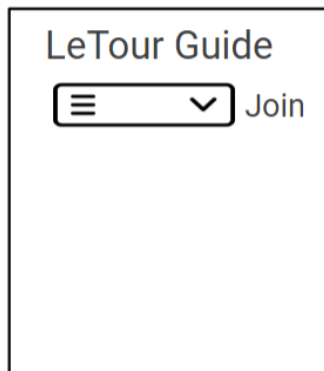
- (3.1.1) Audience homepage
  - (3.1.1a) Tour group selection
  - (3.1.1b) Join button
- (3.1.2) Audience group Tour Page
  - (3.1.2a) Display group name
  - (3.1.2b) uses device speaker
  - (3.1.2c) exit back to home page button (gives warning before action is confirmed)

#### Guide Domain

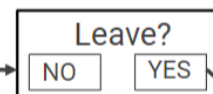
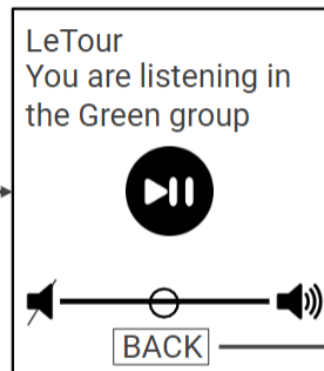
- (3.1.3) Guide homepage
  - (3.1.3a) Text input for Tour group name
  - (3.1.3b) Create group button
- (3.1.4) Guide group Tour Page
  - (3.1.4a) Display group name
  - (3.1.4b) uses device microphone
  - (3.1.4c) exit back to home page button (gives warning before action is confirmed)

## Client Access

host.com/letour



host.com/letour/group/2

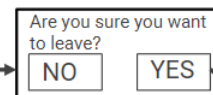


## Guide Access

host.com/guide



host.com/guide/&lt;group name&gt;



## 3.2 Functional Requirements

### Front/Client facing

Front/Client facing requirements are requirements that can be seen by the user. This includes all front-end requirements.

#### 3.2.1 *Landing Page For Audience*

The landing page for the audience members should be accessed by an easily readable and typeable URL. This page presents the user with verification that they have landed at the correct web page such as a large title and short description. This page also shows a selection box for the user to select a tour to join. This might also include a Join button, or some other intuitive way to select a tour.

#### 3.2.2 *In-Tour Page for Audience*

Once the user has selected and “joined” a tour, they will be brought to the “In-Tour” page. This page will show the user their selected tour in large text. It will also contain a button to mute or unmute the audio, as well as including an audio gain slider. Optionally, this page can display the number of connected audience members in that tour. The page will also have a “Leave” button, which will prompt the user to ask for verification, and upon a “Yes” selection or equivalent, take them back to the Landing Page for Audience.

#### 3.2.3 *Landing Page for Tour Guides*

The landing page for tour guides should be accessed by another more specific and hidden URL. This page presents the user with verification that they have landed at the correct web page with a large title, and a subtitle which indicates that they are accessing the guide landing page. A text box will be provided which is the input for the tour group name to create. A button laid beside or under this text box will create the group and direct the user to the In-Tour Page for Tour Guides.

#### 3.2.4 *In-Tour Page for Tour Guides*

Once the user has selected the “Create” button on the Landing Page for Tour Guides, they will be presented with the In-Tour Page for Tour Guides. This page, like the In-Tour Page for Audience, will show the user the tour group name. It will also present a large microphone symbol, followed by the text “MICROPHONE IS ON”. A button on the bottom labeled “Leave” will prompt the user for verification, and upon a “Yes” selection, take them back to the Landing Page for Tour Guides.

### Back facing

Back facing requirements are requirements that are not seen by the user. These are under-the-hood requirements.

#### 3.2.5 *Tour Guide Voice Propagation to Audience*

This function is simple in description but much more difficult in practice. This function will take advantage of a backend web server of some kind (e.g. WebRTC) which provides support for streaming audio. The backend server must be free and open-source. Ideally, this ready-made solution will be easily modifiable or have a simple pluggable module to adapt to this project solution.



---

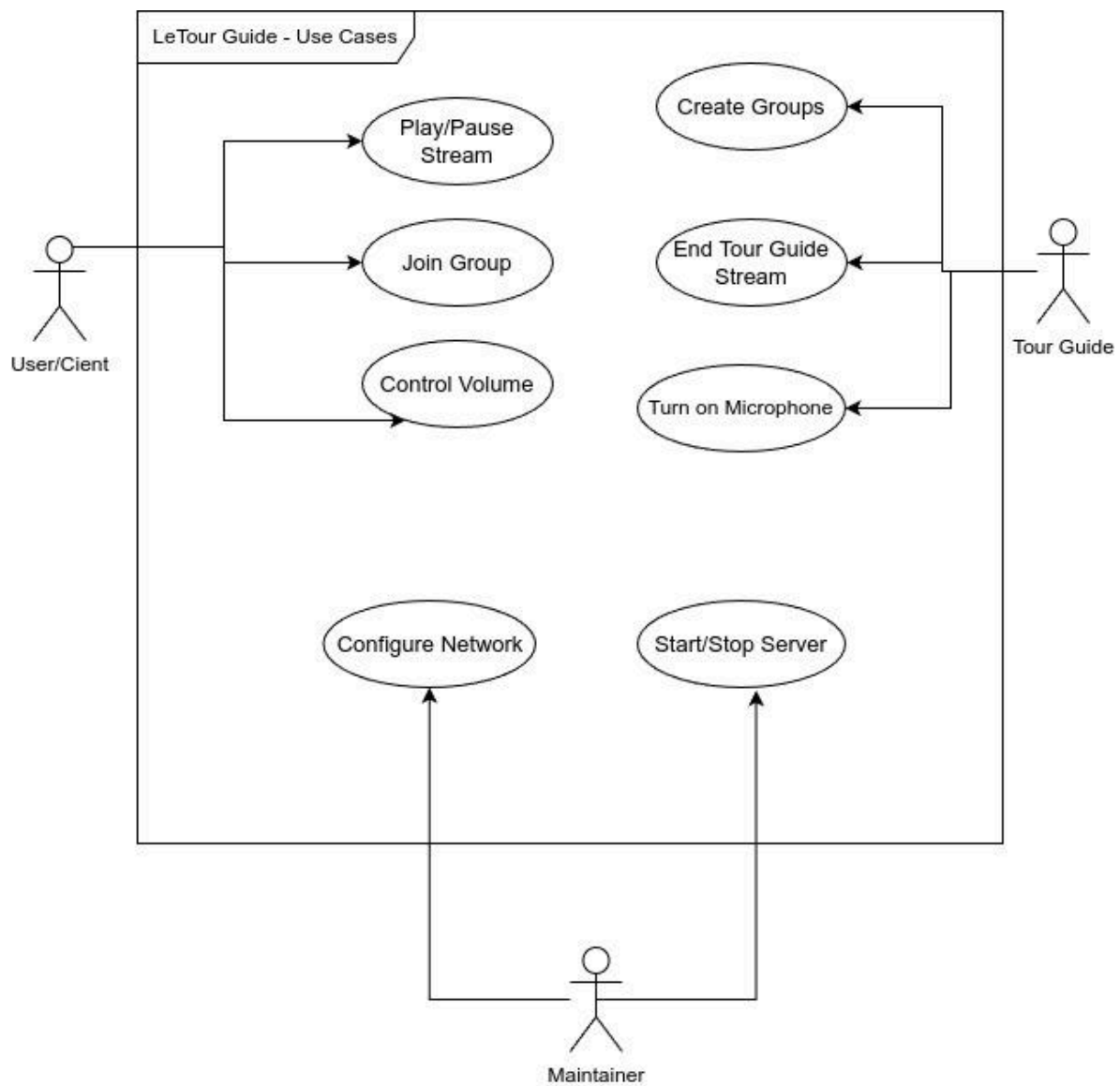
**Optional**

Optional requirements are not critical to the system, but may still be implemented if considerable extra time is found during project development due to unknown areas of acceleration.

**3.2.6 *Tour Questions from Audience***

This optional requirement adds an additional layer of complexity on top of the existing backbone solution. On the tour guide side, a toggle button will allow or disallow “opening the mic” for audience members to ask questions. On the audience side, audience members will now have a button to “take the mic”, which will change the audio input source from the tour guide to that audience member. Once the audience member is done talking, they can press that same button to “hand back the mic” to the tour guide, who can answer the question. Only one client (audience member or tour guide) can speak at a time, so audio mixing will not be needed. The tour guide will also have the ability to manually “take back the mic” if needed. Once question time is finished, the tour guide can disable “taking the mic” by audience members until they activate it again.

### 3.3 Use Case Diagram



## 4 Non-functional Requirements

### 4.1 Software Quality Attributes

#### 4.1.1 Availability

The software need not be constantly online, and in-fact, this is discouraged. The software, up starting up, will remain on unless one of the following actions have happened: the maintainer has shut it down or a module crashes. Upon a crash of the front-end, the back-end should remain online. Upon a crash of the back-end, the front-end should report this to the user in the form of an error page informing the user that the LeTour service is not available.

#### 4.1.2 Portability

Because of the small application of the solution, the software should also have a small footprint. By encapsulating the software in a docker container, the software should remain incredibly portable. The only hindrance to portability is the ability of the server running the container to expose ports to the local network and the local network to recognize the server as a valid server endpoint, however, that is out of the scope of this project.

#### 4.1.3 Correctness

##### 4.1.3.1 Landing Page For Audience

The user can view a list of open tour guides and make a selection. The user can select a tour to join, which brings the user to the In-Tour Page for Audience.

##### 4.1.3.2 In-Tour Page for Audience

The user can hear when the tour guide is speaking. The mute button mutes the tour guide, and displays an image that informs the user that the sound is muted. When the mute button is pressed again, the tour guide can be heard, and the button displays an image that shows that sound is on. The page displays a volume slider which adjusts the gain of the audio from 0% to 100%. A “Leave” button displays a prompt which asks the user if they really want to leave the tour. Upon selecting “No”, they will stay on the page. Upon selecting “Yes”, they will be brought to the Landing Page for Audience.

##### 4.1.3.3 Landing Page for Tour Guides

The user is informed that they are accessing the service as a guide. The textbox accepts alphanumeric characters, and may be of any length. The “Join” button brings the user to the In-Tour Page for Tour Guides. The “Join” button will also create a “room” on the backend server.

##### 4.1.3.4 In-Tour Page for Tour Guides

The user is informed of the group name that they are broadcasting to. The user is presented with the text “MICROPHONE IS ON”. The guide is informed of the number of connected audience members, which updates live when an audience member joins or leaves. A button labeled “Leave” is present. Upon pressing this button, the user is asked if they want to leave. Upon selecting “No”, the

prompt is dismissed. Upon selecting “Yes”, the user is brought back to the Landing Page for Tour Guides. Also, the “room” is destroyed on the backend server.

#### ***4.1.3.5 Tour Guide Voice Propagation to Audience***

Live audio is recorded from only the guide’s microphone, and is audible on all audience devices in the selected room. A tour guide in one room can not be heard by any audience member in another room.

#### ***4.1.3.6 Tour Questions from Audience***

The guide has two buttons on the In-Tour Page for Tour Guides. The first is a toggle button to toggle on and off “taking the mic”. When pressing this button into the “on” position, a “take the mic” button will be visible on each audience member’s In-Tour Page. The second button, which forces the speaker back to the tour guide, is only enabled when an audience member has a speaking role. an audience member takes the speaking role by pressing the “take the mic” button. When this button has been pressed by an audience member, the button is disabled for all other audience members until the speaking role is either given up or forced returned to the guide by the guide. When the guide disables “taking the mic”, the “take the mice” button is removed from all audience member’s displays.

#### **4.1.4 Performance**

The software should be responsive to the user. From the time it takes to accept an HTTP request until its end, the total time elapsed should be not more than one (1) second. Page loading time should take no more than five (5) seconds on a client device.

## **5 Other Requirements**

## Appendix A - Group Log

- 1 Hour (3-4PM)
  - We showed the virtual prototype to Dr. Tixier, and our team discussed an outline for the SRS, distributed tasks, and assigned due dates for the SRS.
- 1430-1530; 1800-1900 Richard Homan sections 1.1 1.2 3.2 4.1
- 1 Hour (9:30-10:30AM)
  - Cameron is working on getting his Docker configured and set up
  - Richard is figuring out how to properly use the GanttProject program to make project schedules
  - Joel managed to create a use case diagram of how the project is supposed to run conceptually.
  - All of us worked on getting the Software Requirements Specification (SRS) document filled out with their respective sections.