Sprint 2

For

TigerTix Development

Prepared by Jonah Colestock and Chris Skelly
Clemson University

Version 0.1

October 28, 2025

Table of Contents

Revision History

Name	Date	Reasons for Change	Version
Jonah Colestock, Chris Skelly	October 28, 2025	Initial version	0.1

Testing

Test Cases

Several unit tests related to the use cases of the website were created. These tests were created using Jest, and include tests for loading the website, using the ticket confirmation, buying a ticket, and using the chatbot to book tickets.
Alongside the unit tests, manual testing was also done for purchasing tickets, using the confirmation system, using the chatbot, and using the speech recognition system.

Expected Results

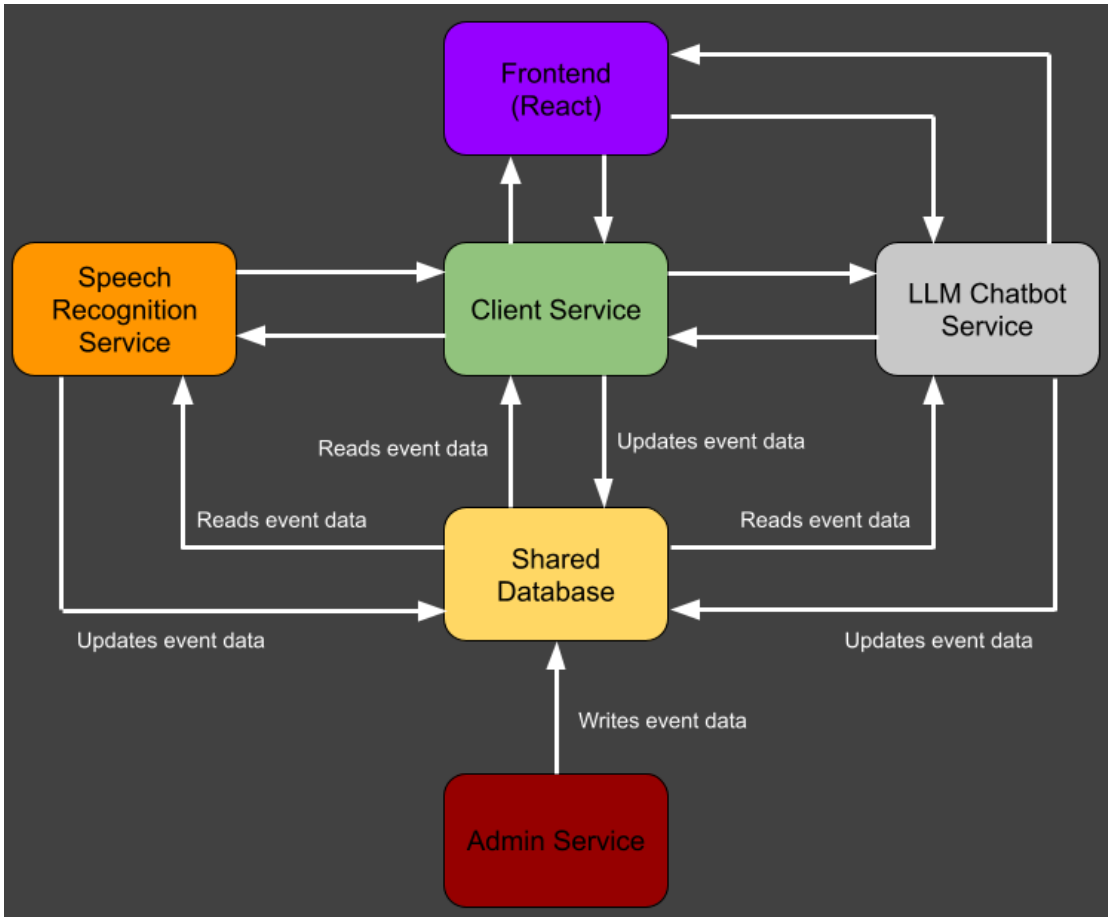
The test for loading the website checks if the button for activating speech recognition is visible to the user. The test for using ticket confirmation simulates a user clicking on one of the “Buy Ticket” buttons and checking if the “Yes” and “No” buttons are then visible. The test for buying a ticket repeats the same process as the previous test, but additionally stores the current ticket amount, simulates buying a ticket, then checks if the ticket amount decreases. The test for the chatbot simulates clicking the chatbot button, entering a prompt, submitting the prompt, then checking if the confirmation appears.

Actual Results

While most of the unit tests were mostly smooth to implement, testing the chatbot functionality using unit testing was a more challenging task because it would repeatedly run into an issue where specific elements were not being recognized by the tests.
Manual testing of the confirmation system helped find a bug where the confirmation would appear on every event when one event was being purchased from.

System Architecture

Architecture Diagram



Architecture Flow

The admin microservice, only available from the backend, fills the shared database with new events and their associated info (name, data, ticket count). When the front end is run, it sends API calls to the client microservice to display the currently-available events and their information. It also sends API calls to the LLM chatbot microservice to give operating power to the chatbot feature. The user can interact with the frontend to buy tickets, which updates the shared database to decrease the amount of tickets available. The frontend is dynamically updated with the current ticket count and the user receives visual confirmation of their ticket purchase. The user can also interact with the chatbot by issuing commands to it, which allows searching tickets on the frontend, or buying tickets (which updates the shared database). Finally, the user can interact with the speech recognition microservice through the frontend, which combines with the LLM chatbot to update the shared database.

Accessibility Description

The front end website application supports the ability to navigate between buttons via pressing the Tab key. This makes it possible to navigate the website without the use of a mouse. Also, a speech recognition service allows for speaking commands without having to type in information, and a text-to-speech service voices text on the screen. Through these combined features, this offers those with difficulties in vision the ability to navigate and use the application.