



Design Use Cases

Team: Tech Support

David Febles - Project Manager

Ayush Shukla - Quality Assurance Lead

Subhash Ramesh - Software Architect

Sabeel Mansuri - User Interface Specialist (I)

Shuusei “Shu” Yoshida - User Interface Specialist (II)

Raghavan Kope - Database Specialist

Somsubhro “Som” Dhar - Business Analyst

Pranav Narasimmaraj - Algorithm Specialist

Kelvin Chan - Senior System Analyst

Amithab Arumugam - Software Development Lead (I)

Nikhil Pathak - Software Development Lead (II)

TABLE OF CONTENTS

Introduction

Legends	4
Glossary	5

Account Details

AD #1 Account Creation	6
AD #2 User Log In	8
AD #3 User Log Out	10
AD #4 Forgot Password	11
AD #5 Change Password	13

Searching

SE #1 General Search	15
SE #2 Search by Department	17
SE #3 Search by Course Number	19
SE #4 Search by Professor	21
SE #5 Search by Quarter	23
SE #6 View All Query Results	25
SE #7 View a Podcast Result's Page	26

Transcripts

TR #1 View Full Transcript for One Podcast	27
TR #2 Podcast Transcript Highlights	29
TR #3 Download Transcript	31

Profile Actions

PA #1 Add a Favorite Podcast	32
PA #2 Unfavorite a Podcast	33
PA #3 View History	35
PA #4 Clear History	36
PA #5 View Favorites	38

Legends

Priority

High: *Vital* - Features that are absolutely essential.

Medium: *Desired* - Features that are highly wanted but not required.

Low: *Optional* - Features that are nonessential add-ons.

Status

Cancelled

Not Started

In Progress

Complete

Glossary

Term	Definition
Account	Email and Password pair used to identify an account
Podcast	A video and audio recording of a specific course lecture
Favorite	A podcast that is specially marked as “Favorite” for easier access later
Transcript	A podcast transcribed to text
Query	A specific key term or phrase that is being searched by the user
Query history	Searches performed in the past tied to an account
Results	List of podcasts that contain the user query with noted timestamps and other relevant podcast information
Podcast Page	Represents the full view page of a single podcast if the user selects a result
Filters	Narrow search by podcast quarter, professor, department, and/or course number
Text-Blurb	Surrounding words spoken in a podcast around the given query
Guest user	A user that is not logged in

AD #1 Account Creation

Priority	Low
Status	Completed
Description	The user creates a new account with an email & password
User Goal	The user wishes to make an account to store and secure personal information
Desired Outcome	The user has successfully created a secure, personal account
Actor	User of the application
Dependent Use Cases	N/A
Requirements	SR #1
Pre-conditions	An account with the same email does not exist and user is not logged in
Post-conditions	An account with the email and password is stored in the database
Trigger	The user wishes to create an account so that they can securely save query history and favorite podcasts to their individual account
Workflow	<ol style="list-style-type: none"> 1) The user shall click on the “SIGN UP” button on the top navbar 2) The frontend shall render the “Signup.js” popup 3) The user shall input their desired email, password, and confirmation password 4) The user shall click the “SIGN UP” button 5) The backend shall validate the email and password 6) The backend shall create an account and store it in the database 7) The frontend shall redirect the user to the “Home.js” page
Alternate Workflow	Invalid Email

	<ul style="list-style-type: none"> 3) The user shall input an invalid or already used email 4) The frontend shall inform them the email is invalid
	<p>Password Mismatch</p> <ul style="list-style-type: none"> 3) The user shall enter a different password in the confirmation box than the other password box 4) The frontend shall inform them the passwords do not match

AD #2 User Log In

Priority	Low
Status	Complete
Description	The user logs into their previously created account to view previous queries and favorite podcasts
User Goal	The user wishes to see information specific to their account stored from previous sessions on the app
Desired Outcome	The user successfully logs in into their account and now can access their account history and favorite podcasts
Actor	User of the application
Dependent Use Cases	AD #1
Requirements	SR #2
Pre-conditions	The user has already made an account and is not logged in
Post-conditions	The user has access to account history and favorites;
Trigger	The user wishes to login in order to view their account history and favorite podcasts
Workflow	<ol style="list-style-type: none"> 1) The user shall click the “LOG IN” button 2) The frontend shall render the “Login.js” popup form 3) The user shall enter their email and password 4) The frontend shall check if the email is of valid form 5) The frontend shall send a POST request of the form to the backend API to verify that the email is valid 6) The backend shall verify the details with the database 7) The frontend shall render the “Home.js” page
Alternate Workflow	Invalid Email/Password <ol style="list-style-type: none"> 2) The user shall input an invalid email/password combination 3) The frontend shall check that the email is of the correct

format

- 4) The frontend shall send a POST request of the form to the backend API to verify that the email is valid**
- 5) The backend shall send a response that this combination does not exist in the database and returns an error to the frontend**
- 6) The frontend shall render the “Login.js” popup with an error message saying “Invalid email/password”**

AD #3 User Log Out

Priority	Low
Status	Complete
Description	The user logs out of their account
User Goal	The user wishes to remove their access to account-specific information and thereby protect their account information
Desired Outcome	The user successfully logs out
Actor	User of the application
Dependent Use Cases	AD #1, AD #2
Requirements	SR #3
Pre-conditions	The user has an account and is logged in
Post-conditions	The user does not have access to account history and favorites
Trigger	The user wishes to protect their privacy by removing access to the account
Workflow	<ol style="list-style-type: none"> 1) The user shall hover over “MY ACCOUNT” in the “Navbar.js” navigation bar 2) The user shall click on the “LOG OUT” button 3) The frontend shall clear the user’s session 4) The frontend shall render the “Home.js” page with the “Navbar.js” navigation bar showing only the option for a logged out user
Alternate Workflow	N/A

AD #4 Forgot Password

Priority	Low
Status	Completed
Description	The user creates a new password to replace the old account password while still maintaining the account history
User Goal	The user wishes to change their account password without needing to remember their previous one
Desired Outcome	The user successfully has a new password for their account
Actor	User of the application
Dependent Use Cases	AD #1
Requirements	SR #4
Pre-conditions	The user has already created an account and is logged out
Post-conditions	The system updates the database to reflect the new password for their account and should be logged out
Trigger	The user wishes to change their password for any reason including increased security or forgotten password
Workflow	<ol style="list-style-type: none"> 1) The user shall click on the “LOG IN” button 2) The frontend shall render the “Login.js” popup form 3) The user shall click on the “Forgot your password?” button 4) The frontend shall render the “Forgotpass.js” page 5) The user shall enter their email 6) The user shall click on the “RESET PASSWORD” button 7) The frontend shall POST the email info to the backend 8) The backend shall verify the email is valid 9) The backend shall send an email to the user’s email with a temporary password recovery token 10) The user shall receive an email with a password recovery token

	<ul style="list-style-type: none"> 11) The user shall input the correct password recovery token & their new, desired & valid password into the form displayed by the frontend 12) The frontend shall POST the user-provided information to the backend 13) The backend shall verify that the password token is correct 14) The backend shall change the user account's password to the newly provided one 15) The frontend shall redirect the user to the "Home.js" page
Alternate Workflow	<p>Invalid email</p> <ul style="list-style-type: none"> 5) The frontend shall display an error message indicating no account exists under that email 6) The frontend shall ask again for user account email 7) Resume at step 5 of expected workflow <p>Incorrect password recovery token</p> <ul style="list-style-type: none"> 11) The user shall input an incorrect password recovery token 12) The frontend shall POST the user-provided information to the backend 13) The backend shall validate the information & reject the reset request, due to an incorrect recovery token 14) The frontend shall display an error to the user specifying that the password recovery token is incorrect

AD #5 Change Password

Priority	Low
Status	Completed
Description	The user creates a new password to replace the old account password while still maintaining the account history
User Goal	The user wishes to change their account password while remembering their previous one
Desired Outcome	The user successfully has a new password for their account
Actor	User of the application
Dependent Use Cases	AD #1, AD #2
Requirements	SR #5
Pre-conditions	The user has already created an account and is logged in
Post-conditions	The system updates the database to reflect the new password for their account
Trigger	The user wishes to change their password for any reason including increased security
Workflow	<ol style="list-style-type: none"> 1) The user shall hover over “MY ACCOUNT” 2) The user shall click on the “SETTINGS” button 3) The frontend shall render the “Changepass.js” page 4) The user shall enter the correct previous password, a valid new password, and a verification of the new password 5) The frontend will confirm that the format for the password is valid 6) The frontend shall POST the form to the backend. 7) The backend shall change the password in the database 8) The frontend shall display a success message for password change
Alternate Workflow	Incorrect previous password

- | | |
|--|--|
| | <ul style="list-style-type: none">5) The frontend shall present an error message indicating the previous password was entered incorrectly6) Resume at step 4 of expected workflow |
|--|--|

Password Mismatch

- | | |
|--|--|
| | <ul style="list-style-type: none">5) The frontend shall present an error message indicating the new password and password confirmation were different6) Resume at step 4 of expected workflow |
|--|--|

SE #1 General Search

Priority	High
Status	Complete
Description	The user queries for keywords/terms without any filters
User Goal	The user wishes to find a keyword/term in any available podcast from any department, course, professor, or quarter
Desired Outcome	The user sees unfiltered results from every podcast available in our application
Actor	User of the application
Dependent Use Cases	N/A
Requirements	SR #6
Pre-conditions	The user has not entered any filters to search with and is on either the “Home.js” page or “Results.js” results page
Post-conditions	The user receives unfiltered results for their query in all podcasts the application services
Trigger	The user wants to query all classes independent of any filters to find relevant discussions to their query
Workflow	<ol style="list-style-type: none"> 1) The frontend shall render the “Home.js” page which also renders the “Search.js” component, rendering the search bar 2) The guest user shall enter a query into the search bar 3) The guest user shall click the “SEARCH” button or <ENTER> 4) The frontend shall send a GET request with the query to the backend 5) The backend shall run the query in the database and send back relevant results 6) The front end shall render the “Results.js” page with all the results

Alternate Workflow	<p>The system has no results for the given query</p> <ul style="list-style-type: none">6) The frontend shall render the “Results.js” page and display a “No results found.” message for the user query <p>The user is logged in</p> <ul style="list-style-type: none">5) The backend shall add the user’s query to the user’s history6) The backend shall run the query in the database and send back relevant results7) The front end shall render the “Results.js” page with all the results
---------------------------	--

SE #2 Search by Department

Priority	High
Status	Complete
Description	The user queries for a term in a specific department like CSE
User Goal	The user wishes to narrow search results to just classes in the specified department
Desired Outcome	The user only sees results from the department they want
Actor	User of the application
Dependent Use Cases	SE #1
Requirements	SR #7
Pre-conditions	The user has not specified any department to filter the search, and is on either the “Home.js” page or “Results.js” results page
Post-conditions	The user receives results for their query in the department they specified
Trigger	The user wishes to narrow their search results to one department to get a specific perspective
Workflow	<ol style="list-style-type: none"> 1) The guest user shall enter a query into the search bar 2) The frontend shall render the different filters on the page 3) The guest user shall enter the department they are interested in via the filter bars 4) The guest user shall click the “SEARCH” button 5) The frontend shall send a GET request with the form information to the backend 6) The backend shall run the query in the database and send back the results 7) The frontend shall render the “Results.js” page with the relevant results

Alternate Workflow	<p>The user shall input a department where the keyword does not appear</p> <ul style="list-style-type: none">7) The frontend shall render the “Results.js” page and display a “No results found.” message for the user query <p>The user is logged in</p> <ul style="list-style-type: none">6) The backend shall add the user’s query to the user’s history7) The backend shall run the query in the database and send back relevant results8) The front end shall render the “Results.js” page with all the results
---------------------------	--

SE #3 Search by Course Number

Priority	High
Status	Complete
Description	The user searches for a term in a specific class like CSE 21
User Goal	The user wishes to narrow search results to just a specific class
Desired Outcome	The user only sees results from the class they want
Actor	User of the application
Dependent Use Cases	SE #1
Requirements	SR #7
Pre-conditions	The user has not specified any class to filter the search and is on either the “Home.js” page or “Results.js” results page
Post-conditions	The user receives results from their query in the class they specified
Trigger	The user wishes to narrow their search results to one class to get a specific perspective of the topic at hand
Workflow	<ol style="list-style-type: none"> 1) The guest user shall enter a query into the search bar 2) The frontend shall render the different filters on the page 3) The guest user shall enter a course number they are interested in 4) The frontend shall enable the course number filter and render relevant options 5) The guest user shall enter a course number 6) The guest user shall click the “SEARCH” button 7) The frontend shall a GET request with the form information to the backend 8) The backend shall run the query in the database and send back the results 9) The frontend shall render the “Results.js” page with the relevant results

Alternate Workflow	<p>The user shall input a course number where the keyword does not appear</p> <ul style="list-style-type: none">4) The frontend shall render the “Results.js” page and display a “No results found.” message for the user query <p>The user is logged in</p> <ul style="list-style-type: none">8) The backend shall add the user’s query to the user’s history9) The backend shall run the query in the database and send back relevant results10) The front end shall render the “Results.js” page with all the results
---------------------------	--

SE #4 Search by Professor

Priority	High
Status	Complete
Description	The user searches for a term said by a specific professor
User Goal	The user wishes to narrow search results to just a specific professor
Desired Outcome	The user only sees results from the professor they want
Actor	User of the application
Dependent Use Cases	SE #1
Requirements	SR #7
Pre-conditions	The user has not specified any professor to filter the search and is on either the “Home.js” page or “Results.js” results page
Post-conditions	The user receives results from their query for the professor they specified
Trigger	The user wishes to narrow their search results to one professor to get a specific perspective of the topic at hand
Workflow	<ol style="list-style-type: none"> 1) The guest user shall enter a query into the search bar 2) The frontend shall render the different filters on the page 3) The guest user shall enter a professor they are interested in 4) The guest user shall click the “SEARCH” button 5) The frontend shall a GET request with the form information to the backend 6) The backend shall run the query in the database and send back the results 7) The frontend shall render the “Results.js” page with the relevant results

Alternate Workflow	<p>The user shall input a professor who does not speak about the keyword in any podcast</p> <ul style="list-style-type: none">7) The frontend shall render the “Results.js” page and display a “No results found.” message for the user query <p>The user is logged in</p> <ul style="list-style-type: none">6) The backend shall add the user’s query to the user’s history7) The backend shall run the query in the database and send back relevant results8) The front end shall render the “Results.js” page with all the results
---------------------------	---

SE #5 Search by Quarter

Priority	High
Status	In Progress
Description	The user searches for a term said during a specific quarter
User Goal	The user wishes to narrow search results to just a specific quarter
Desired Outcome	The user only sees results from the quarter they want
Actor	User of the application
Dependent Use Cases	SE #1
Requirements	SR #7
Pre-conditions	The user has not entered any quarter to filter the search and is on either the “Home.js” page or “Results.js” results page
Post-conditions	The user receives results from their query for the quarter they specified
Trigger	The user wishes to narrow their search results to one quarter to get a specific perspective of the topic at hand
Workflow	<ol style="list-style-type: none"> 1) The guest user shall enter a query into the search bar 2) The frontend shall render the different filters on the page 3) The guest user shall enter a quarter they are interested in 4) The guest user shall press the “SEARCH” button 5) The frontend shall send a GET request with the form information to the backend 6) The backend shall run the query in the database and send back the results 7) The frontend shall render the “Results.js” page with the relevant results

Alternate Workflow	<p>The user shall input a quarter where the keyword does not appear</p> <ol style="list-style-type: none">7) The frontend shall render the “Results.js” page and display a “No results found.” message for the user query <p>The user is logged in</p> <ol style="list-style-type: none">6) The backend shall add the user’s query to the user’s history7) The backend shall run the query in the database and send back relevant results8) The front end shall render the “Results.js” page with all the results
---------------------------	---

SE #6 View All Query Results

Priority	High
Status	Complete
Description	After the user queries, they will see a list results of all relating podcasts that contain the user query, and these results will display a text-blurb and the name of the course, professor, and lecture number of each podcast result
User Goal	The user wishes to see a list of all relating podcasts pertaining to their query
Desired Outcome	The user sees a list of podcast results which contain the user query
Actor	User of the application
Dependent Use Cases	SE #1
Requirements	SR #8, SR #9
Pre-conditions	The user has searched for a keyword using any or none of the filters
Post-conditions	The user receives a page of at most 10 podcast results sorted by their relevance to the query
Trigger	The user wishes to view podcasts that contain their query
Workflow	<ol style="list-style-type: none"> 1) The frontend shall render the “Results.js” page with the top relevant results retrieved from the backend 2) The user shall view all the podcast results with corresponding text-blurbs & timestamps given after the search
Alternate Workflow	<p>No results found</p> <ol style="list-style-type: none"> 1) The frontend shall render the “Results.js” page with message “No results found.” displayed

SE #7 View a Podcast Result's Page

Priority	High
Status	Complete
Description	After the user queries and is given a list of results, they will click a single result to view the information of that podcast
User Goal	The user wishes to gain more information about a particular result podcast
Desired Outcome	The user sees a short transcript of the part of the lecture containing their keyword(s), has access to view the podcast video at the appropriate time-stamp where the query was discussed, and has access to view the full transcript of the lecture
Actor	User of the application
Dependent Use Cases	SE #1, SE #6
Requirements	SR #9, SR #10, SR #11, SR #12, SR #13
Pre-conditions	The user has searched for a keyword using any or none of the filters and is given a list of podcast results
Post-conditions	The user receives a page with a short text-blurb containing the query, the podcast moved to the appropriate timestamp, and transcripts available below
Trigger	The user wishes to view a single podcast result at the timestamp where their query was discussed
Workflow	<ol style="list-style-type: none"> 1) The user shall click the result they wish to view 2) The frontend shall render the respective "PodcastPage.js" page with the timestamped podcast, text blurb, and view full transcript option, for the result selected
Alternate Workflow	N/A

TR #1 View Full Transcript for One Podcast

Priority	High
Status	Complete
Description	After clicking a single result for the query, the user can view the full transcript for the entire podcast on the individual results page
User Goal	The user wishes to see the transcript for the entire podcast for any reason, including lack of headphones in a quiet setting or enhanced learning experience through reading lectures
Desired Outcome	A full transcript of the podcast is displayed in a paragraph form
Actor	User of the application
Dependent Use Cases	SE #1, SE #6, SE #7
Requirements	SR #11
Pre-conditions	The user has entered a query, seen the list of results, and has clicked one of the results to view an individual podcast page
Post-conditions	The user receives a full transcript of the selected podcast in paragraph form on the podcast page
Trigger	Without needing to watch the podcast at the appropriate timestamp, the user would like the full text of the podcast that contains the query to gain more context
Workflow	<ol style="list-style-type: none"> 1) The user shall click the “View Full Transcript” link on the individual podcast page 2) The frontend shall send a GET request for the transcript to the backend 3) The backend shall send the transcript to the frontend 4) The frontend shall render the “Transcript.js” page with the full podcast transcript

Alternate Workflow	N/A
---------------------------	------------

TR #2 Podcast Transcript Highlights

Priority	Low
Status	Cancelled
Description	After clicking a single result for the query, the user can view the full transcript for the entire podcast on the results page which will contain highlighted words that matched the query
User Goal	The user wishes to view the context around the search query within the podcast's transcript while still knowing where the query results were located
Desired Outcome	A full transcript of the podcast is displayed in a paragraph form with the queries highlighted
Actor	User of the application
Dependent Use Cases	SE #1, SE #6, SE #7
Requirements	SR #15
Pre-conditions	The user has entered a query, seen the list of results, and has clicked one of the results to view an individual podcast page
Post-conditions	The user receives a full transcript of the selected podcast in paragraph form on the podcast page with highlights of the original query
Trigger	The user would like to view the context around their query without having to watch the podcast
Workflow	<ol style="list-style-type: none"> 1) The user shall click the "View Highlighted Transcript" button on the individual podcast page 2) The frontend shall send a GET request for the transcript to the backend 3) The backend shall query the database and return a transcript for the podcast back 4) The frontend shall display the full highlighted transcript for the podcast the user has clicked on in the results page

Alternate Workflow	N/A
---------------------------	------------

TR #3 Download Transcript

Priority	Low
Status	Cancelled
Description	After clicking a single result for the query, the user can download the full transcript for the entire podcast
User Goal	The user wishes to obtain a transcript of a podcast outside of the application
Desired Outcome	The user is able to download the transcript of a podcast onto their system
Actor	User of the application
Dependent Use Cases	SE #1, SE #6, SE #7
Requirements	SR #16
Pre-conditions	The user has entered a query, seen the list of results, and has clicked one of the results to view an individual podcast page
Post-conditions	The user receives the full transcript on their system for the podcast via a download
Trigger	Without needing to watch the podcast at the appropriate timestamp, the user would like to locally download the full text of the podcast that contains the query to gain more context and save the transcripts for use without needing the application
Workflow	<ol style="list-style-type: none"> 1) The user shall click the “View Full Transcript” button on the individual podcast page 2) The frontend shall display the “Transcript.js” page for the podcast the user has clicked on in the results page 3) The user shall click on the “Download” button 4) The frontend shall download the podcast transcript onto the user’s device
Alternate Workflow	N/A

PA #1 Add a Favorite Podcast

Priority	Low
Status	Complete
Description	The user saves a podcast to their account's favorite list
User Goal	The user wishes to add a podcast to their account's favorite list
Desired Outcome	The podcast is added to the user account's favorite list
Actor	User of the application
Dependent Use Cases	AD #1, AD #2, SE #1, SE #6, SE #7
Requirements	SR #13
Pre-conditions	The user has an account, is logged in, has entered a query, and has selected a specific podcast result
Post-conditions	The podcast is within the user account's favorite list
Trigger	The user finds that a specific podcast is important and wishes to add it to their favorites list
Workflow	<ol style="list-style-type: none"> 1) The user shall click the "FAVORITE" button 2) The frontend shall POST a request to the backend to add the current podcast to the user's list of favorites 3) The backend shall add the podcast to the user's favorites list in the database 4) The frontend shall display a pop-up notification that indicates that the podcast has been added to the favorites list
Alternate Workflow	N/A

PA #2 Unfavorite a Podcast

Priority	Low
Status	Complete
Description	The user removes a podcast from their account's favorite list
User Goal	The user wishes to remove a podcast from their account's favorite list
Desired Outcome	The podcast is removed from the user account's favorite list
Actor	User of the application
Dependent Use Cases	AD #1, AD #2, SE #1, SE #6, SE #7, PA #1
Requirements	SR #13
Pre-conditions	The user has an account, is logged in, and has already favorited a podcast in the past
Post-conditions	The podcast is no longer within the user account's favorite list
Trigger	The user no longer finds that a specific podcast is important and wishes to remove it from their favorites list
Workflow	<ol style="list-style-type: none"> 1) The user shall hover over "MY ACCOUNT" 2) The frontend shall render a dropdown menu underneath "MY ACCOUNT" 3) The user shall click the "FAVORITES" button 4) The frontend shall render the "Favorites.js" page 5) The user shall click the star icon next to the podcast they want to remove 6) The frontend shall send a DELETE request to the backend with the podcast to remove from the user's list of favorites 7) The backend shall remove the podcast from the user's favorites list in the database 8) The frontend shall re-render the "Favorites.js" page to no longer display the unfavorited podcast 9) The frontend shall display a pop-up that indicates the

	podcast has been removed from the favorites list
Alternate Workflow	Unfavoriting from the Podcast Page <ol style="list-style-type: none"> 1) The user shall click the “UNFAVORITE” button at the bottom of podcast page 2) The frontend shall send a DELETE request to the backend with the podcast to remove from the user’s list of favorites 3) The backend shall remove the podcast from the user’s favorites list in the database 4) The frontend shall display a pop-up that indicates the podcast has been removed from the favorites list

PA #3 View History

Priority	Medium
Status	Complete
Description	The user views a list of all of their previous queries and the podcasts they have watched in the past
User Goal	The user wishes to see what queries they have entered and what podcasts they have viewed in the past
Desired Outcome	The user sees the history related to their account
Actor	User of the application
Dependent Use Cases	AD #1, AD #2, SE #1, SE #6, SE #7
Requirements	SR #14
Pre-conditions	The user has an account and is logged in
Post-conditions	The user receives a list of their search history and viewed podcast history
Trigger	The user wants to view their history to see what they viewed or queried in the past in case they want to review material
Workflow	<ol style="list-style-type: none"> 1) The user shall hover over “MY ACCOUNT” 2) The frontend shall render a dropdown menu underneath “MY ACCOUNT” 3) The user shall click the “HISTORY” button 4) The frontend shall send a GET request to the backend to get the users history 5) The frontend shall render the “History.js” page with the user’s search and view history
Alternate Workflow	N/A

PA #4 Clear History

Priority	Medium
Status	Completed
Description	After the user deletes their search history, their search history is cleared
User Goal	The user wishes to clear their search history for any reason, including starting a new quarter with a new history
Desired Outcome	The user can no longer view previous queries or podcasts watched
Actor	User of the application
Dependent Use Cases	AD #1, AD #2, SE #1, SE #6, SE #7
Requirements	SR #17
Pre-conditions	The user has an account and is logged in
Post-conditions	The user's search history is empty
Trigger	The user wants to clean their search history to avoid viewing previous queries
Workflow	<ol style="list-style-type: none"> 1) The user shall hover over "MY ACCOUNT" 2) The frontend shall render a dropdown menu underneath "MY ACCOUNT" 3) The user shall click the "HISTORY" button 4) The frontend shall render the "History.js" page 5) The user shall click the "CLEAR HISTORY" button 6) The frontend shall send a DELETE request to the backend to clear the user's history 7) The backend shall delete the user's history from the database 8) The frontend shall re-render the "History.js" page with all its field cleared 9) The frontend shall display a pop-up indicating the history has been cleared

Alternate Workflow	N/A
---------------------------	------------

PA #5 View Favorites

Priority	Medium
Status	Complete
Description	The user views a list of all of their previous favorited podcasts
User Goal	The user wishes to see what podcasts they have favorited in the past
Desired Outcome	The user sees the past favorited podcasts from their account
Actor	User of the application
Dependent Use Cases	AD #1, AD #2, SE #1, SE #6, SE #7, PA #1
Requirements	SR #18
Pre-conditions	The user has an account and is logged in
Post-conditions	The user receives a list of their favorited podcasts
Trigger	The user wants to view their favorite podcasts from the past so they can see the podcasts they have deemed important to revisit
Workflow	<ol style="list-style-type: none"> 1) The user shall hover over “MY ACCOUNT” 2) The frontend shall render a dropdown menu underneath “MY ACCOUNT” 3) The user shall click the “FAVORITES” button 4) The frontend shall send a GET request to the backend to get the users favorites 5) The frontend shall render the “Favorites.js” page with the user’s favorite podcasts
Alternate Workflow	N/A