Draft Term Project Report

Apple Podcast Improvement

Elisa Simpson, Katherine Skarda, Charles Penn

IT-210-A

Marymount University

Table Of Contents

Introduction/Purpose:

The Apple Podcast developed by Apple Inc., allows users to explore, subscribe, and listen to podcasts. This application is compatible with other Apple products such as Macbooks, iPads, iPhones, and Apple Watches. Users have access to various podcasts libraries regarding a wide range of topics. These topics range from news, to education, and even storytelling. The interface allows users to search and browse podcasts all while keeping a tab open for any podcast that is currently playing. The application offers other features such as the ability to create your own library. The app has no "stop playing" feature, and will automatically start whenever connected to car bluetooth which is distracting and could be hazardous. Our idea of adding a stop button would improve both the app's functionality and the user's experience. The button would look and function similar to the YouTube app's "X" at the right side of the player controls when the video is in MiniPlayer mode. This would allow the user to clear a podcast from staying active in the background, which would keep the app from unexpectedly starting it up when the user doesn’t want that happening.

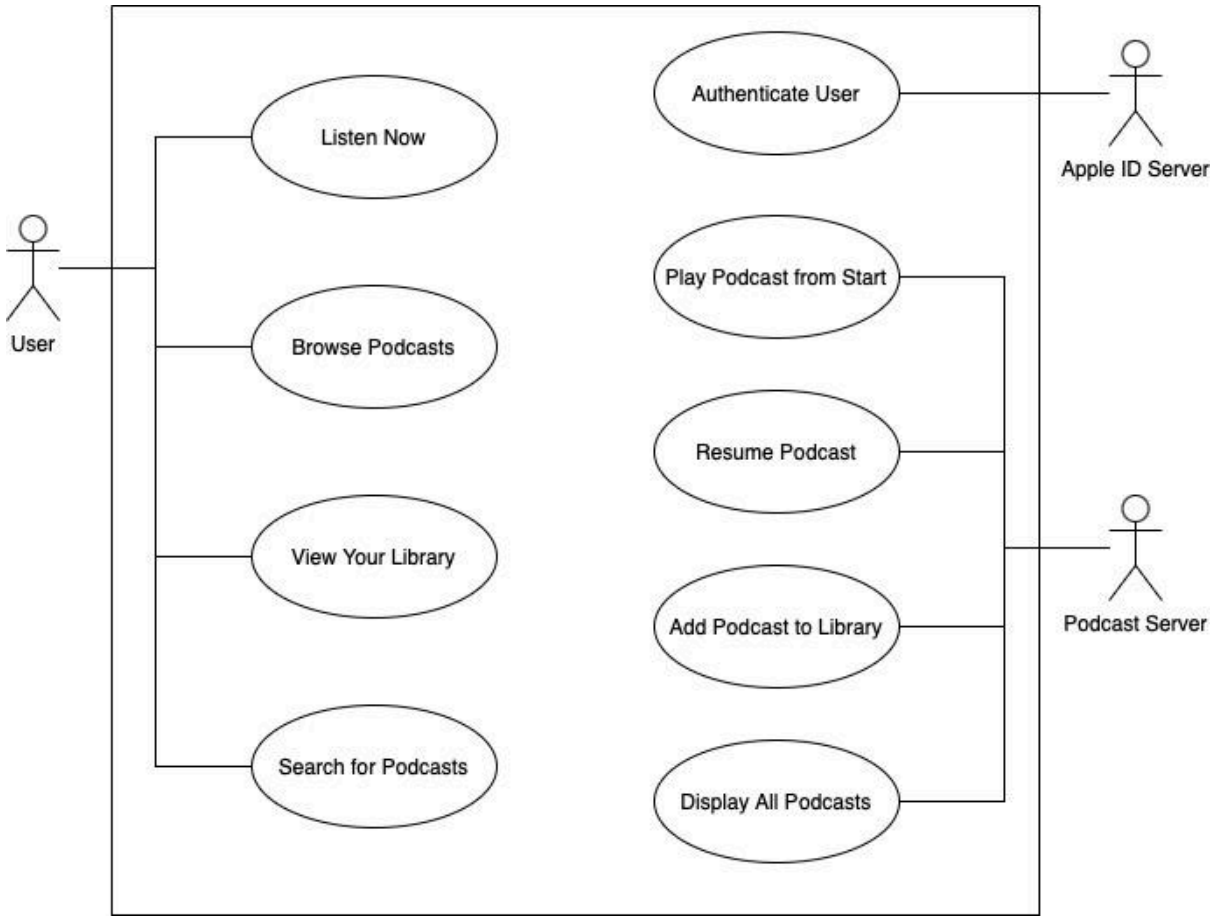
System Requirements:

Non-Functional: The non-functional requirements for the Apple Podcast focuses more on the systems operations. For the Apple Podcast, the application must have good performance which will allow the users to access and listen to the podcast without interruptions or buffering. The app must be compatible with other Apple products. For example, If a car has Apple CarPlay, users should be able to access and control the application similarly to how they would on their mobile devices. The app should have high scalability to allow a large number to use the app

without degrading the app's performance. The problem we want to fix is the app's usability. Exiting the podcast or quitting the app should stop the podcast without any extra effort from the user.

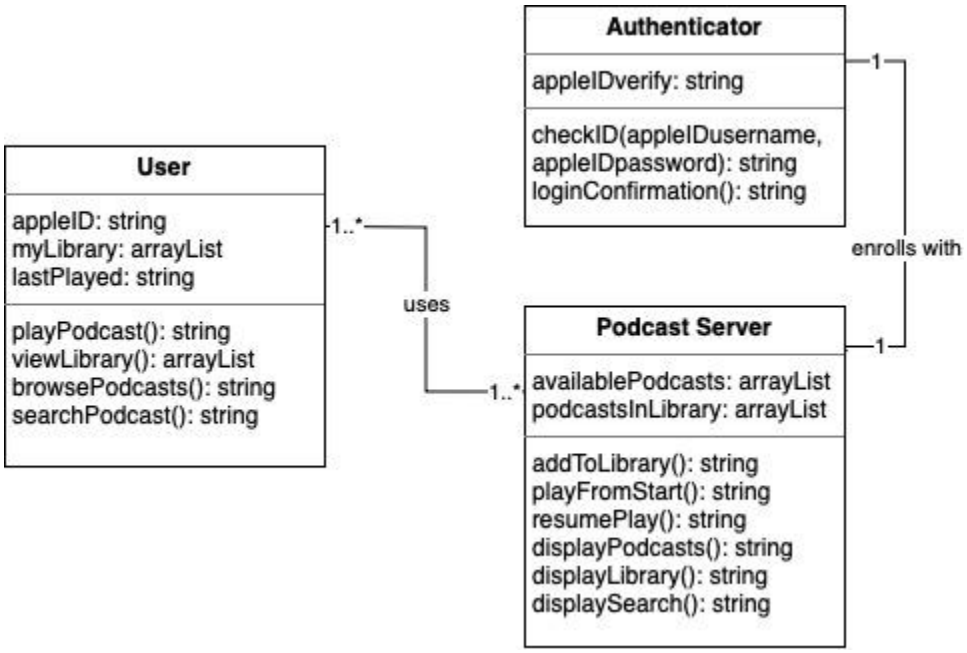
Functional: These are requirements that the user requests that are mandatory. They define the product features and what the product must accomplish. For the Apple Podcast app it must authenticate users using their Apple ID. This ensures that the person using the app is the valid owner of the device which is imperative for security purposes. The app must also allow users to browse and search for podcasts. This is the whole idea behind the app; a space where people can post and find different podcasts that interest them to listen to. An option to save podcasts to libraries must be included in the app as well. This builds off of the browse function. Once the user finds a podcast they are interested in then instead of having to listen to them immediately they can save it to their library to listen to at a later date. Once the user chooses the podcast that they want to listen to, the app must play the podcast with the designated output device. Finally the app must allow the user to pause the podcast, fast forward, rewind, and stop it at their own discretion.

Use Case Diagram with Description:



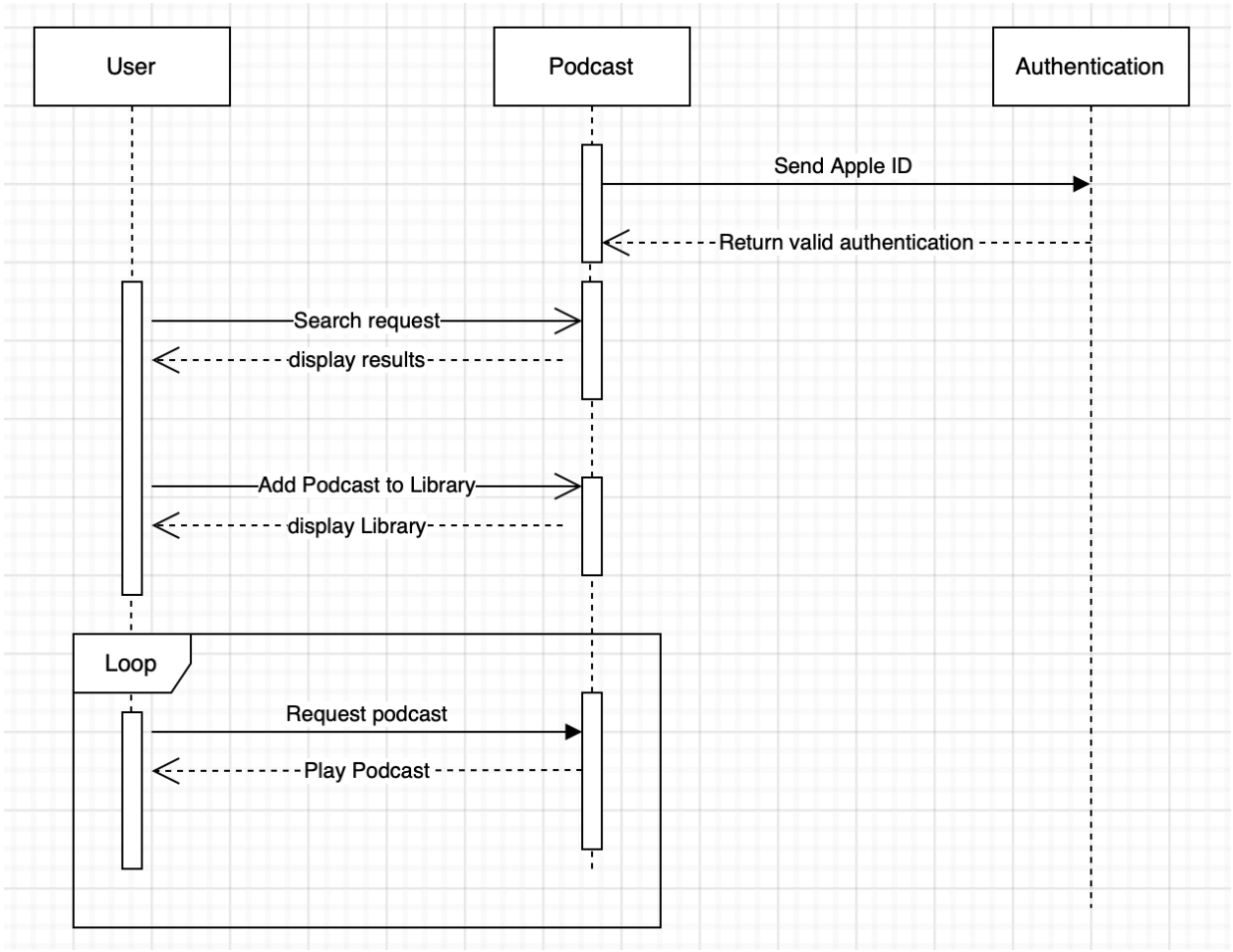
In the use case diagram for Apple Podcast, the primary actor is the application user, and the secondary actors are the application server and the ID server. When the user opens Apple Podcast, their credentials are automatically checked by the Apple ID server. After they are recognized the user is brought to the home page where they can choose to listen to a podcast, browse all podcasts, view their personal library, or search for a specific podcast. Depending on which path the user decides to take, the podcast server will take the appropriate matching action to produce the desired result. The preconditions are that the user already has a valid Apple ID account and they have downloaded the app. Post conditions on success would be the user finding and smoothly playing the podcast of their choice.

Class Diagram with Description:



We have three classes named User, Authenticator, and Podcast Server. The User class has the attributes appleID and lastPlayed with the data type string and myLibrary with data type arrayList. Example data for these would be johnsmith@icloud.com, Splendid Table Spring Baking, and the collection of names of the podcasts the user has selected. The Authenticator class has the attribute appleIDverify of the data type string, with example data of verified or not verified. The Podcast Server class has attributes availablePodcasts and podcastInLibrary, both of which are data type arrayList. The example data for these would be a collection of podcast names, either all of the podcasts available or the ones that the user selected.

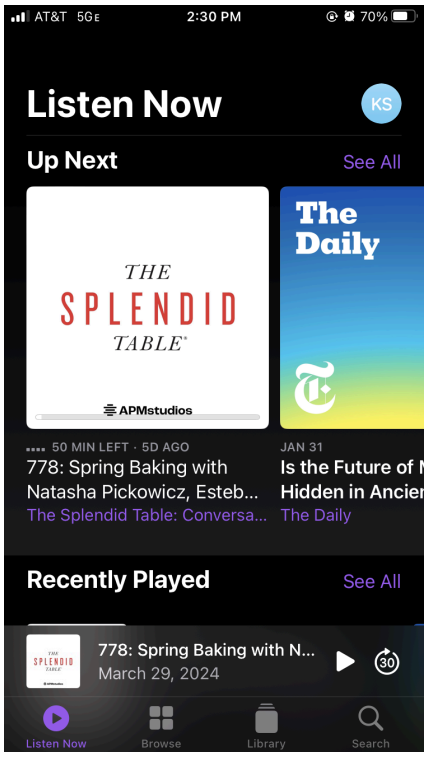
Sequence Diagram with Description:

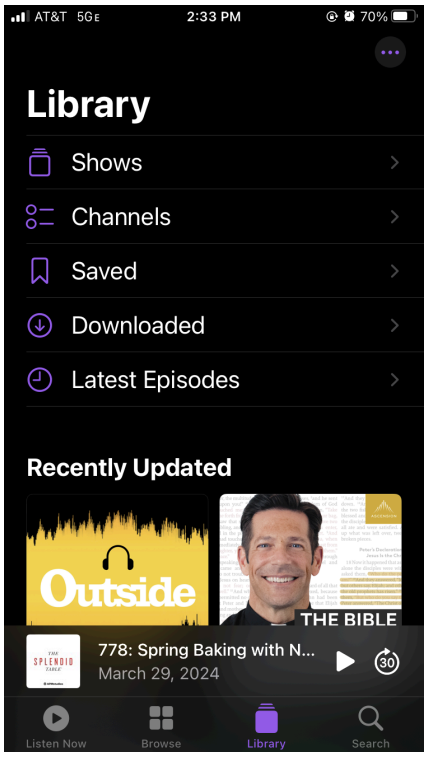


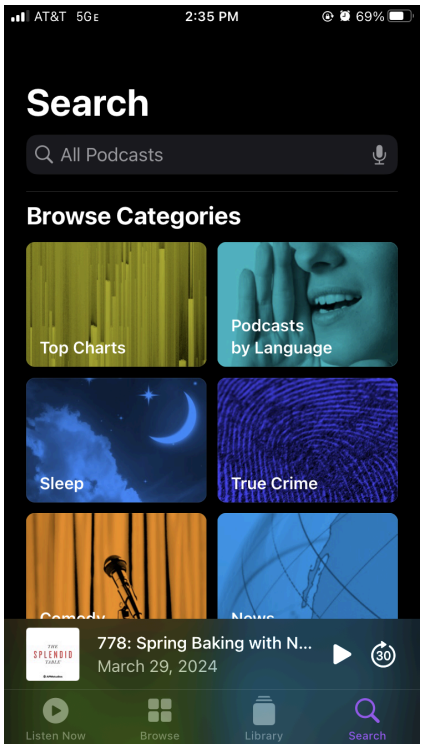
Our sequence diagram has three lifelines, the user, the podcast and then an authenticator. The first activation box sends the Apple ID information to the authenticator and it will wait for a response before continuing. Once the authenticator verifies a valid Apple ID then it will return the authentication and allow the User to access the podcast app. The next activation box is the user requesting a search from the Podcast lifeline. The Podcast lifeline processes the request and returns the search results to the User. After the User has searched for a podcast then they can add that podcast to their library by sending a request to the Podcast lifeline. Once the Podcast receives the request to add a podcast to the library it will carry that task out and then return the

full library in return to the User. After this happens the User can play a podcast by requesting the specific one they want to play from the Podcast lifeline. The Podcast lifeline will retrieve the podcast from the server and then return it to the user. This action will be looped since the app has an auto play function. Once a podcast ends then the next one will play immediately so this loop will only end when the user pauses it or closes out of the app.

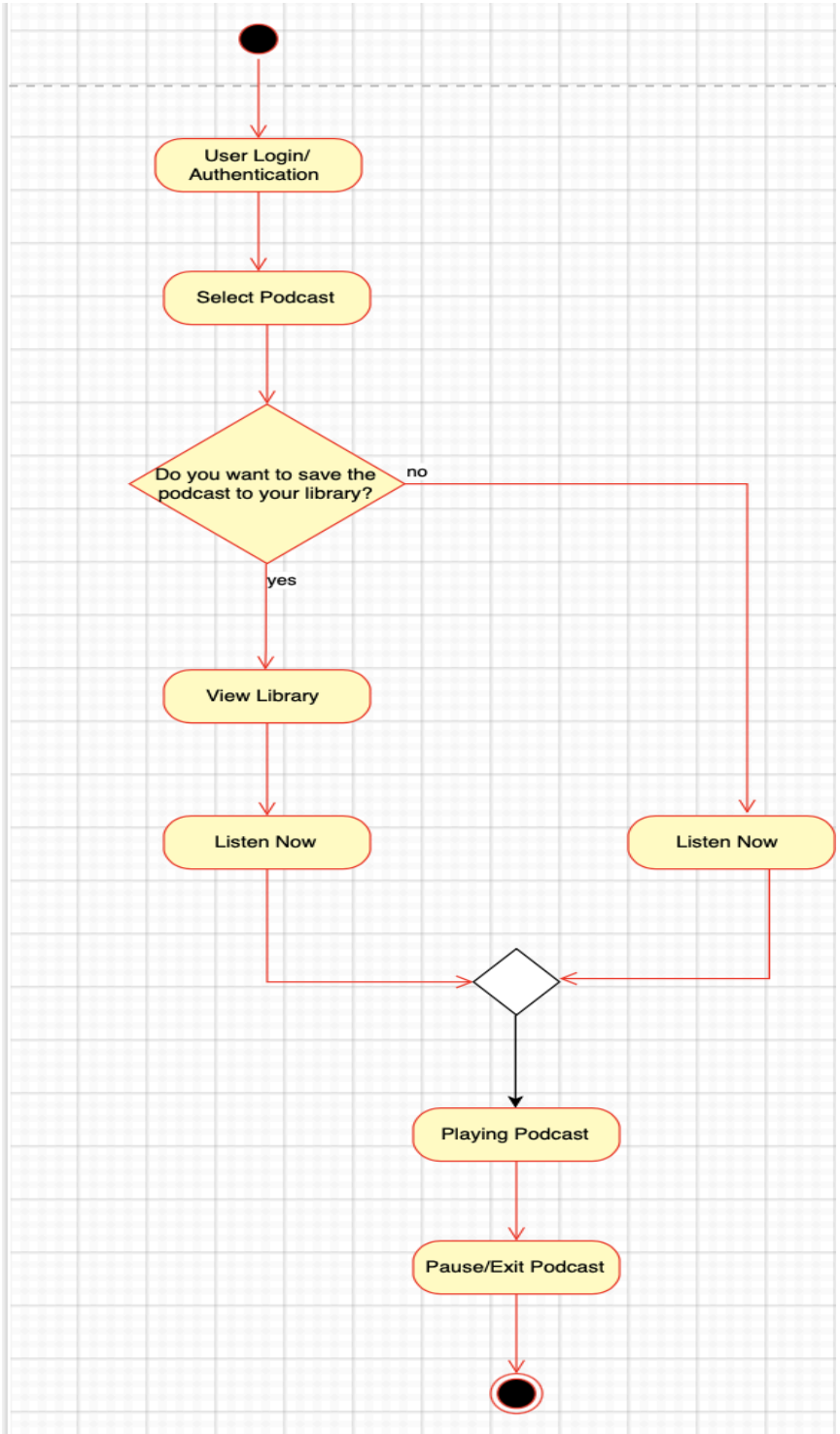
UX/UI:


AT&T 5GE
2:30 PM
70%
Listen Now
KS
Up Next
See All
THE SPLENDID TABLE
APMstudios
The Daily
50 MIN LEFT · 5D AGO
778: Spring Baking with Natasha Pickowicz, Esteb...
The Splendid Table: Conversa...
JAN 31
Is the Future of
Hidden in Ancie
The Daily
Recently Played
See All
778: Spring Baking with N...
March 29, 2024
Listen Now
Browse
Library
Search


AT&T 5GE
2:33 PM
70%
Library
Shows
Channels
Saved
Downloaded
Latest Episodes
Recently Updated
Outside
THE BIBLE
778: Spring Baking with N...
March 29, 2024
Listen Now
Browse
Library
Search

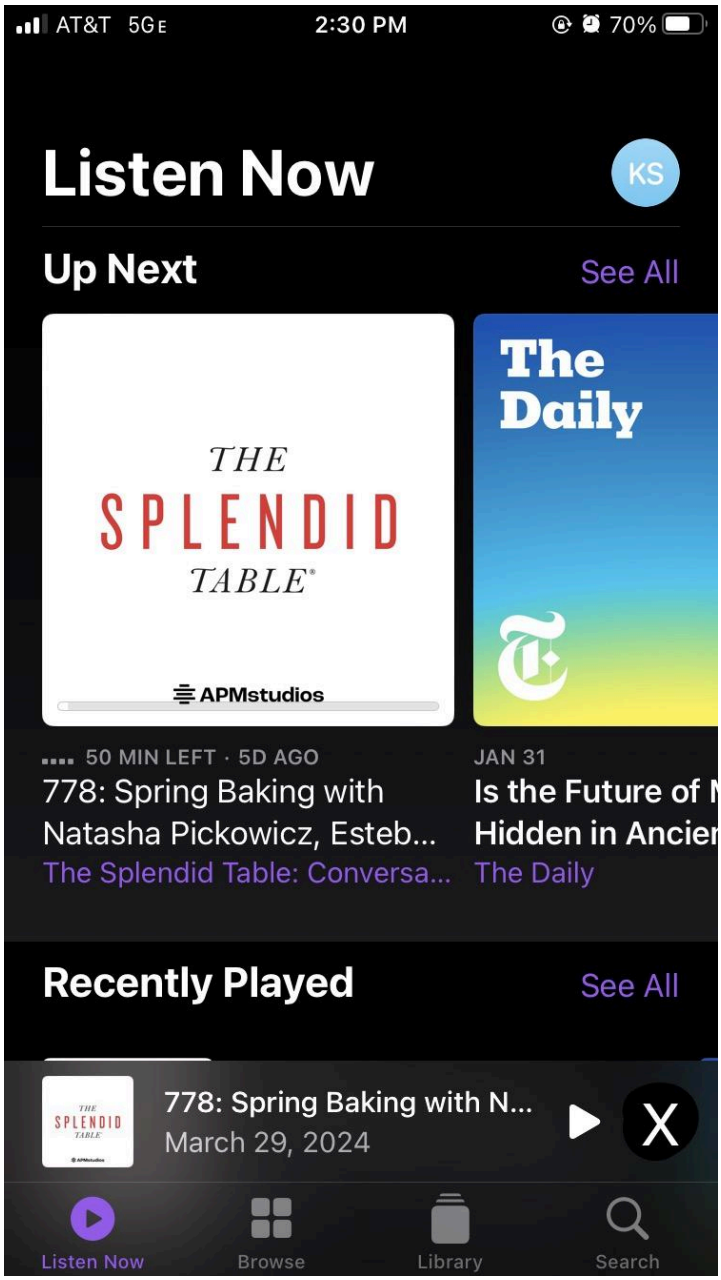

AT&T 5GE
2:35 PM
69%
Search
All Podcasts
Browse Categories
Top Charts
Podcasts by Language
Sleep
True Crime
778: Spring Baking with N...
March 29, 2024
Listen Now
Browse
Library
Search

Activity Diagram with Description:



In the activity diagram, we wanted to display the flow of events with the resolved paused feature. The diagram starts with the user entering the application and logging in/authenticating their account. The authentication will take place in the authentication server. Once the user is authenticated, they will be able to select a podcast. Once they have made their selection, they will have the option to upload the podcast to their personal library. If they choose to add it to their personal library the screen will display their library to ensure their new podcast was added. The user will then be able to listen to the podcast. However, if the user decides not to add the podcast to their personal library, they will be able to just play the podcast. After starting the podcast, the events merge in the diagram and the application will continue playing the podcast. Once the user pauses or exits the podcast, the podcast will no longer play and the program ends.

Prototype:



The prototype above shows our concept of the Apple Podcast application improved by the addition of a stop playing feature, observed on the lower right of the screen. By clicking this feature, the user would be able to close the mini player banner across the bottom of the screen

and clear whatever is actively playing. This would open more space for browsing on the home screen along with the main purpose of removing it from automatically playing when the user's device connects via bluetooth.

Conclusion:

Apple Podcasts is one of the most used podcasts apps in the United States and provides access to educational, informational, and entertaining content. It works through a straightforward user interface to allow smooth and easy access to podcast listeners. The application meets both the functional and non-functional requirements. However, our prototype shows the simple yet vital addition of a stop playing button. A potential challenge that may occur during the implementation of this improvement would be ensuring the pause button is also fixed in other Apple products and not just mobile devices. For example, if the user has Apple CarPlay and connects to the Apple Podcast, the app stops playing after the user presses pause or exits the car. Overall, this improvement would allow users to clear the queue of currently playing, thus allowing for the equally important requirement in life - quiet time for contemplation.