

# Mini-Player Feature on Campaign Landing Pages

## Glossary

	Terms	Explanation
1	<b>In-app campaign</b>	Campaign Landing Page is rendered when a user clicks on a barker on home screen in android OS Amazon Music Application, user is landed on a page for that campaign for example My Music Moments. The Campaign Page is hosted through web-view.
2	<b>Campaign Deeplink</b>	Deeplink parameter is a URL parameter that is added to a link to direct a user to a specific section or page within an application or website irrespective of from where the user is trying to invoke it(email, whatsapp, etc..).
3	<b>Web-View</b>	A web-view is an Android view that displays web pages within an Android application. It is essentially a mini-browser within the app. In Android via web-view is just like any other web page, except that it is being displayed within an Android app.
4	<b>Barker</b>	Carousal Widget used as an ingress (on Browse Home) to Music campaigns inside the app.

## Documents

**Q/A doc with platform team:** Mini-Player Feature on campaign landing Pages-Q/A

## Overview

This is a new feature where music and podcast mini-player can be overlayed on Campaign Landing Pages. Currently, whenever a campaign landing page is opened on android OS, existing streaming podcast or music tracks were not controllable until the user gets back on home screen. This is because the mini-player is hidden after clicking on the barker on CLP. With this feature, it will enhance the user experience with the current media streaming on the app after opening the campaign page and user will have choice to control the playback from the CLP as well.

We are using URL deeplink parameter in existing campaign deeplinks which would enable the android developers of Amazon Music application to have flexibility of display(enable/disable) of mini-player over Campaign Landing Pages. Either we can use existing info in URL of campaigns or a Deeplink parameter is introduced in existing URL which will be used to have flexibility of display(enable/disable) of mini-player over Campaign Landing Pages.

## Problem Statement

Currently on the Android platform, the mini-player is hidden behind by the landing page (or in general Webview), creating a poor user experience. This issue is not in parity with iOS. Moreover, users are unable to pause or stop the current song as a result. A solution is needed to ensure that the landing page does not interfere with the mini-player and that users have control over their music playback.

## Stakeholders

List out all POCs for the project, including (but not limited to):

- SDE : Siddharth Varshney
- SDM : Shilpa Ramachandra Reddy
- PM : <NA>
- TPM : Ritesh Rai
- UX : <NA>
- QA : Amit Gupta/Chandrappa D R

## Requirements

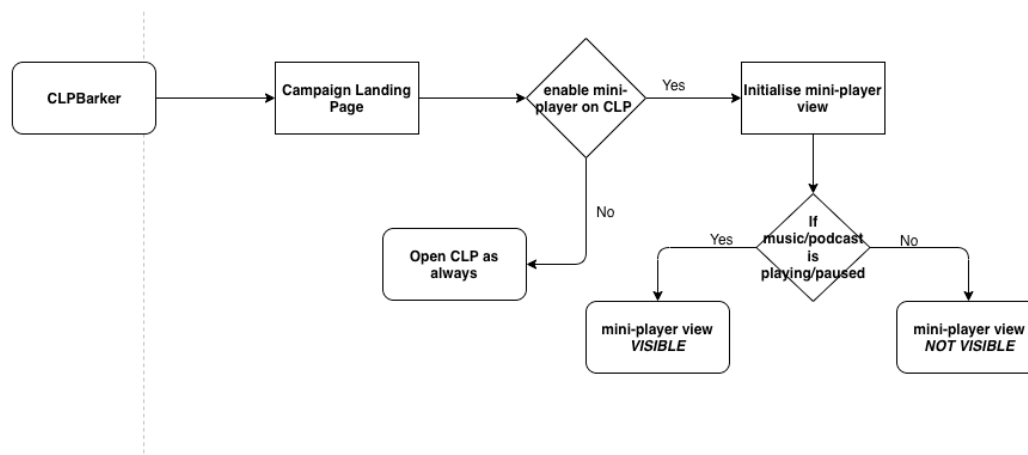
### FUNCTIONAL REQUIREMENTS :

1. **P0** : If in case mini-player is active and user clicks on the barker, then the mini-player should not be overlapped with the campaign landing page. [Applicable both for music and podcast]
2. **P0** : If in case user is playing music on Amazon Music application and user clicks on the deeplink from outside the application then the campaign landing page should be initialised along with the mini-player without any overlap for the customer to track the progress of the song. [Applicable both for music and podcast]
3. **P0** : User should be able to interact with the mini-player without any hinderance. [i.e : play/pause and skip to the next song] [Applicable both for music and podcast]
4. **P0** : The addition of mini-player should be configurable as per the requirement for handling the cases where mini-player need not to be added along with the landing page.
5. **P0** : The complete feature of mini-player along with WebView should be gated with Arcus and Weblab in order handle worst case scenarios.

### NON-FUNCTIONAL REQUIREMENTS :

1. **Latency** - Mini Player overlay on webview shouldn't impact the either Webview load or streaming latency.
2. **Consistency** - The track/podcast streaming progress should be consistent :
  - a. if the user interacts with the mini-player with/without landing page.
  - b. Expansion the mini-player into NPS with/without landing page.
  - c. User clicks back from webview.
3. **Configurability** : The addition of mini-player along with the webview/landing page should be configurable in order to give more flexibility.
4. **Extensibility** : The feature should be extensible to other landing pages as well if in case required in future.

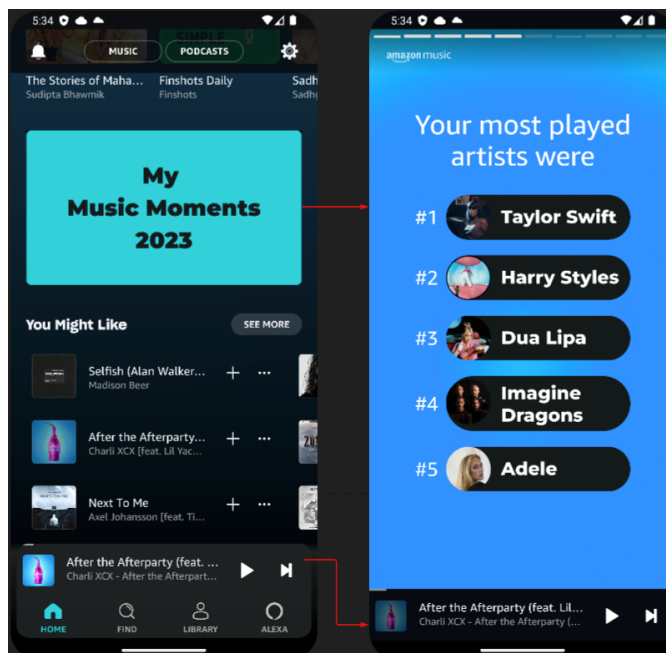
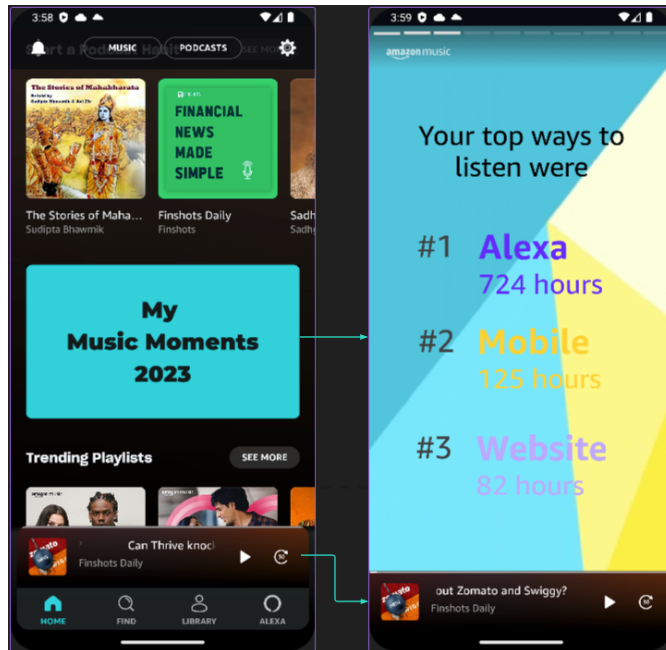
## High Level Flow



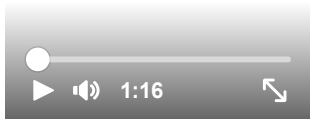
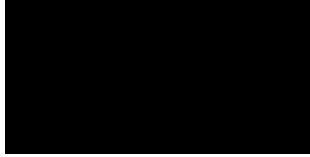
On Android Music application home screen, there are barker of various campaigns. When the user selected a campaign barker the campaign landing page opens up in Music app. Now it's developer's choice to make mini-player enabled on the campaign landing page. If we like to show mini-player on that campaign page, we would introduce a query parameter which would enable it on the Campaign Landing Page.

Music mini-player and Podcast mini-player, both are included in this feature. The mini-player would have equivalent functionality what it has on Home page. Which means that when the user taps on mini-player, user would be routed to Now Playing Screen.

## UI Overview:

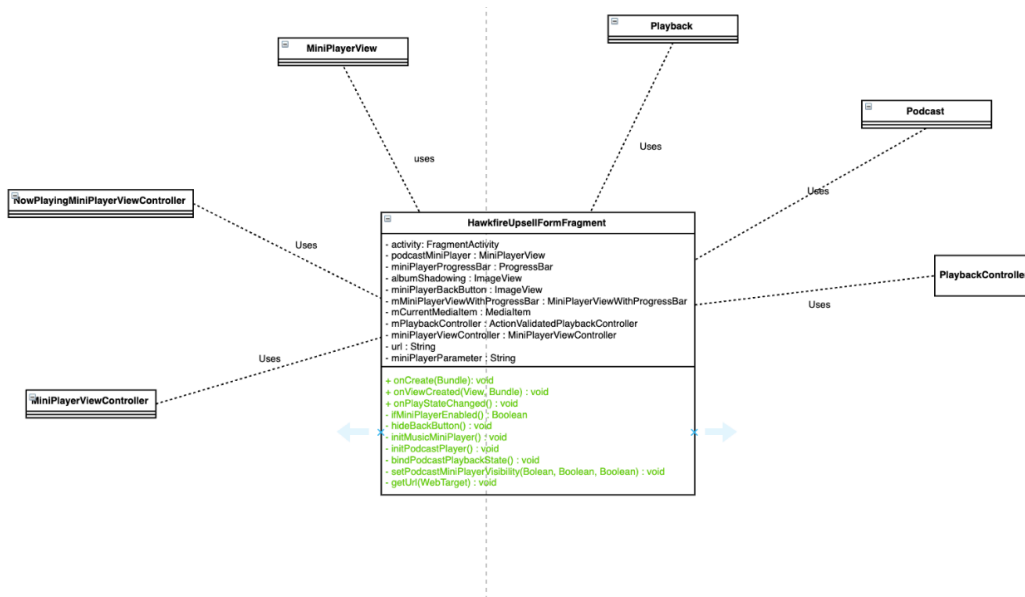


Video for complete functioning of UI:



## Proposed Approach ( DigitalMusicAndroid3P)

Overlay a mini-player on Campaign Landing Page whenever user has a song/podcast in a playing or paused state.



Package affected: DigitalMusicAndroid3P

In DigitalMusicAndroid3P we are working on *HawkfireUpsellFormFragment.java* in which we have all methods which are implemented as new and modified constructor of the class to get URL from web Target(deeplink parameter) to implement mini-player overlay flexibility on campaigns. We have initialised url through `getUrl()` method.

We have proposed two approaches to enable mini-player only on selected(LPN owned) Campaign Landing Pages. We can use existing URL parameter “campaign/invite to enable the mini-player on campaigns who’s deeplink contains this parameter, and second is if a campaign contains “miniplayer=true” in the URL(deeplink) will have mini-player overlay on landing page. The approach mentioned is the only possible approach which could be implementable as:

- Adding the mini-player view on web-view layout is a must, without that, we cannot have interactible mini-player overlay.

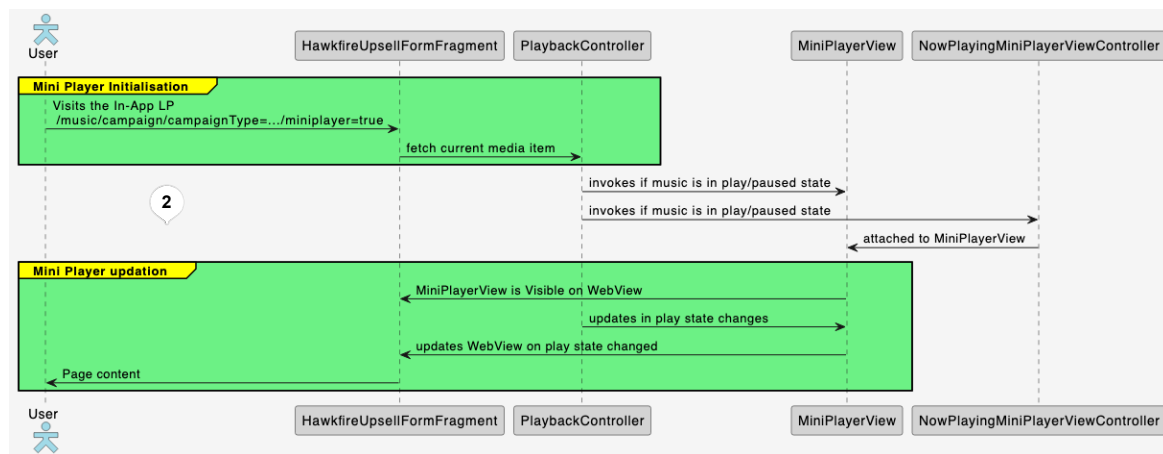
This rolls out the approach of resizing/limiting the web-view to top of mini-player and bottom bar at home screen.

- There were two class which were considered while adding logic to mini-player present on web-view. *BaseWebFragment* and *HawkFireUpsellFragment*.
  - With BaseFragment, all the Campaign Landing pages will have the mini-player overlay globally, while with HawkFireUpsellFragment, the campaigns which are hosted through the class only they will have the overlay.
  - Moreover, the flexibility to disable mini-player overlay on a campaign page would not work on for BaseWebFragment implementation as we are not passing any URL web target to the class a part of which would be used as a parameter to decide the enabling/disabling of mini-player on a landing page.

## Low Level Design

Music mini-player and Podcast mini-player are implemented independent of one another in Music app which necessitates independent implementation in web-view to ensure they function autonomously from one another. As such, it is imperative to design them to operate in parallel without impacting the user experience. This requires careful consideration and the implementation of separate functionalities to ensure that they work seamlessly without hindering each other.

### UML - MUSIC MINI-PLAYER



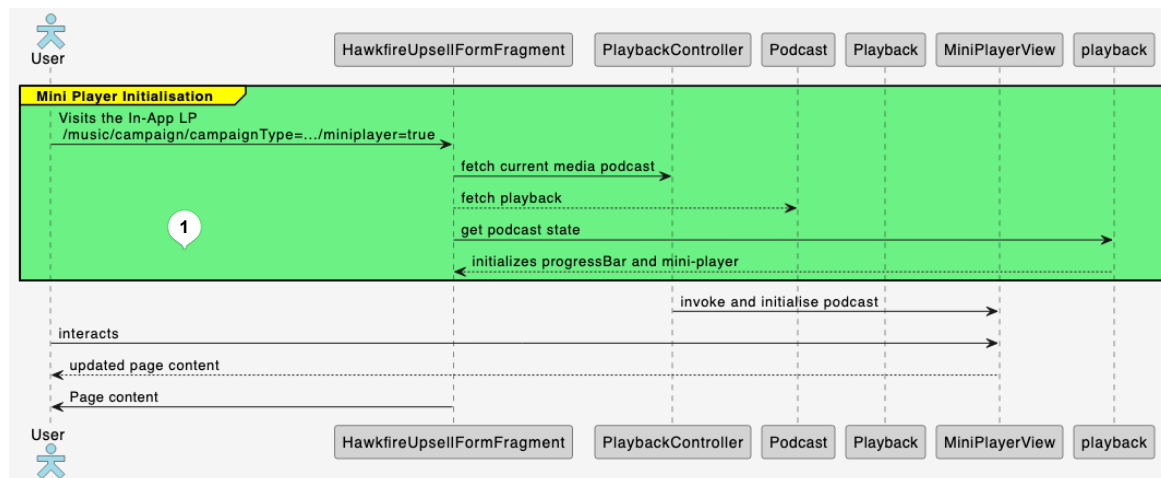
After user clicks on campaign barker on home page, web-view is invoked through *HawkfireUpsellFormFragment.java*. On which *PlaybackController* is invoked which gets current media which is played as well detects if we change the media.

If there is a music media in progress (playing/paused) *MiniPlayerView.java* class is initialised with current media through *NowPlayingMiniPlayerViewController.java* connects view on xml with logic of mini-player view and is rendered visible on campaign screen. If there is no music media in progress, we do not initialise further classes.

Changing the music invokes *onPlayStateChangeListener* which updates *PlaybackController* and mini-player renders correct media.

The logic of other functionalities are handled within *MiniPlayerView* class itself.

### UML - PODCAST MINI-PLAYER



After user clicks on campaign barker on home page, web-view is invoked through *HawkfireUpsellFormFragment.java*. *PlaybackController* is invoked, *Podcast.java*, and *Playback* is invoked through different functions.

From *Podcast* class if we get podcast as media from *podcast.getPlayback()*, we would show the podcast mini-player according to states for podcast for example *BUFFERING*, *PLAYING*, *STOPPED*.

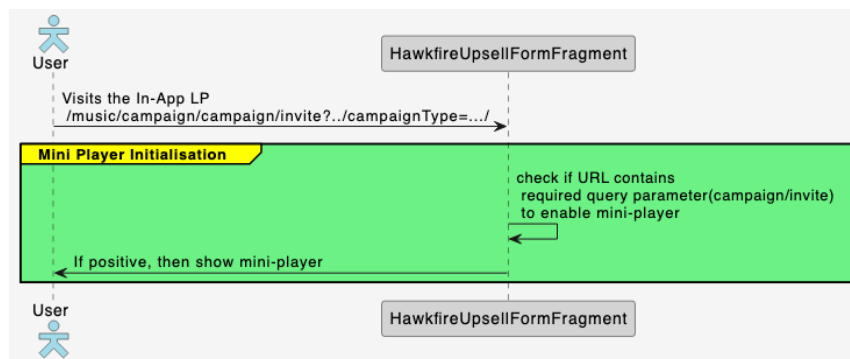
## UX CHANGE (XML FILE)

Add mini-player and progress bar in Web-view *FrameLayout*.

## DEEPLINK UML (FLAG TO RESTRICT MINI-PLAYER ONLY TO LPN OWNED CAMPAIGNS)

### APPROACH 1 (RECOMMENDED)

To enable mini-player player on any campaign landing page hosted by LPN team, we can add flag that if URL contains “campaign/invite” in deeplink query parameter, only then we would enable the mini-player. As all the campaigns which contains “campaign/invite” are hosted by LPN team, we could target mini-player only on those campaign landing pages.



## Current Behaviour

While we are introducing the endpoint to have flexibility over display of mini-player over any campaign, after inclusion of this feature it would not affect the current behaviour of any existing campaign. As the existing deeplinks for all current active campaigns which is not owned by LPN does not have the endpoint “campaign/invite”, mini-player would not be visible on those pages.

## Pros

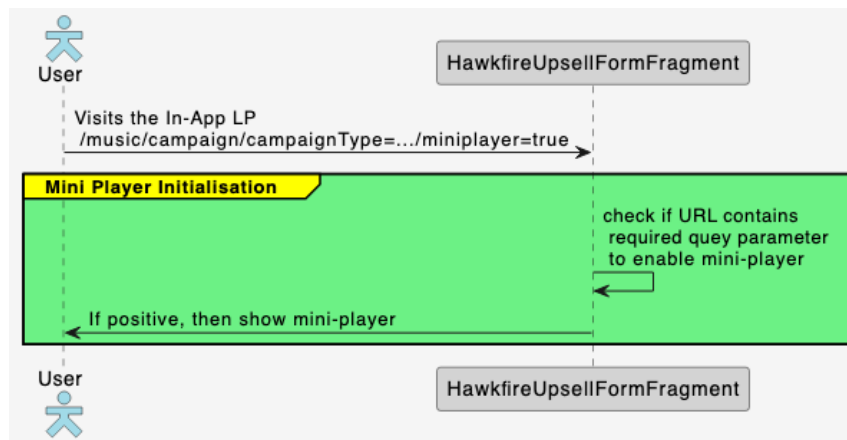
- No parameter needs to be added in deeplinks which already exists for existing campaigns, saving time and effort.
- Able to target campaigns which are owned only by LPN without affecting current behaviour of other existing campaigns.

#### Cons

- In future if deeplink URI changes it would require change to handle the new endpoint for campaign landing pages.

#### APPROACH 2

To enable the mini-player on any campaign page, the developer would add an extra query parameter. If the query parameter is present in campaign deeplink, the mini-player would be rendered on the campaign page.  
proposed QueryParam: "miniplayer=true".



#### Current behaviour

While we are introducing the query parameter to have flexibility over display of mini-player over any campaign, after inclusion of this feature it would not affect the current behaviour of any existing campaign. As the existing deeplinks for all current active campaigns does not have the query parameter "miniplayer=true", mini-player would not be visible on those pages.

#### Pros

- As long as the URL contains our desired parameter, even if the URL changes, it would not affect the behaviour of mini-player.
- With extra URL parameter, any team would be able to enable mini-player on their campaign as long as they add desired parameter to their campaign deeplink URL.

#### Cons

- A parameter(miniplayer=true) needs to be added in all the existing deeplink URL of campaigns where we need to enable the mini-player.
- Adding extra parameter in all the existing campaign deeplinks might take time and effort.

#### RECOMMENDATION

We are going ahead with the first approach

#### APPROVALS

	Date	Status
Shilpa Ramachandra Reddy		
Ritesh Rai		
Divashree Agrawal		Approved
Shubham Guglani		
Ravi Kumar		
LPN Tech team		
Swamalakshmi Rengarajan		

## REFERENCES:

The reference have been taken from codes from screens which are already implementing music-mini-player decoupled from Home screen and Podcast-mini-player implemented on home screen.

- [X-Ray page mini-player](#)
- [Now Playing Mini-player on Home Screen](#)
- [Lyrics Page mini-player](#)