User Management Transition System

X (states):

1. Adminpage
2. Userloginpage
3. Userpage

X_0 (initial states):

1. Adminpage
2. Userloginpage

U (actions):

1. Create_user
2. Update_user
3. userlogin
4. logout
5. viewusers

F (transition relation):

State	Action	NextState	Obseravtion
Adminpage	Create_user	Adminpage	Directed to page containing form to create new user
Adminpage	Update_user	Adminpage	A form to modify details of user is shown
Userloginpage	userlogin	Userpage	Userpage is displayed for login into application
Userpage	logout	Userloginpage	The userloginpage is shown, when logged out of application.
Adminpage	viewusers	Adminpage	A page containing list of users will be shown

Y (observables):

1. Directed to page containing form to create new user
2. A form to modify details of user is shown
3. User’s login screen is displayed for login into application
4. User logout of application
5. A page containing list of users will be shown

h (displaymap):

1. h(Adminpage)=A page containing all functionalities required for user management is shown
2. h(Userpage)= User's login screen is displayed for login into application
3. h(Userloginpage)= The userloginpage is shown, when logged out of application

Library Management

X (states):

1. Userpage
2. Usermusicaddpage
3. Musicformpage
4. MymusicPage
5. Nowplayingpage

X_0 (initial states):

1. Userpage

U (actions):

1. View_Music
2. Add_music
3. Add_imported_music
4. Add_fromExternalSource
5. Upload_music
6. View_my_music
7. View_albums
8. View_artists
9. View_LatestAlbums
10. View_MostListenedAlbums
11. Search_music
12. Search_playlist
13. View_nowplaying
14. Play_track
15. Play_nextTrack
16. Play_previousTrack
17. Shuffle
18. Repeat_playlist
19. Like_track
20. Unlike_track

F (transition relation):

State	Action	Nextsate	Obseravtion
Userpage	View_Music	Userpage	The page displays the list of music
Userpage	Add_music	Usermusicaddpage	Page containing the options to upload music is shown.
Usermusicaddpage	Add_imported_music	Musicformpage	A form to upload the music is shown
Usermusicaddpage	Add_fromExternalSource	Musicformpage	A form to upload the music from external source is shown.
Usermusicaddpage	Upload_music	Musicformpage	An option to upload music file from is shown
Userpage	View_my_music	MymusicPage	Navigated to Mymusictab to list all music
Mymusicpage	View_albums	Mymusicpage	List of all Albums are shown
Mymusicpage	View_artists	Mymusicpage	List of all Artists are shown
Userpage	View_LatestAlbums	Mymusicpage	Navigated to Mymusicpage to show the latest albums
Userpage	View_MostListenedAlbums	Mymusicpage	Navigated to Mymusicpage to show the Most listened albums
Userpage	Search_music	Mymusicpage	Displays the Mymusictab that containing the music if it is there, as specified in the search query.
Userpage	Search_playlist	Userpage	Displays the playlist if it is there, as specified in the search query
Userpage	View_nowplaying	Nowplayingpage	Displays the current music being played
Userpage	Play_track	Userpage	Track will be played
Userpage	Play_nextTrack	Userpage	Next track in the playlist will be played

Userpage	Play_previousTrack	Userpage	Previously played track will be played
Userpage	Shuffle	Userpage	Shuffles the order of tracks in the playlist
Userpage	Repeat_playlist	Userpage	The previously played playlist will be added to the queue for playback.
Userpage	Like_track	Userpage	A red-heart is shown for the track liked
Userpage	Unlike_track	Userpage	A white-heart is displayed for the liked track when it is unliked

Y(observables):

1. The page displays the list of music
2. Page containing the options to upload music is shown.
3. A form to upload the music is shown
4. A form to upload the music from external source is shown.
5. An option to upload music file from is shown
6. Navigated to Mymusictab to list all music
7. List of all Albums are shown
8. List of all Artists are shown
9. Navigated to Mymusicpage to show the latest albums
10. Navigated to Mymusicpage to show the Most listened albums
11. Displays the Mymusictab that containing the music if it is there, as specified in the search query.
12. Displays the playlist if it is there, as specified in the search query
13. Displays the current music being played
14. Track will be played
15. Next track in the playlist will be played
16. Previously played track will be played
17. Shuffles the order of tracks in the playlist
18. The previously played playlist will be added to the queue for playback.
19. A red-heart is shown for the track liked
20. A white-heart is displayed for the liked track when it is unliked

h (displaymap):

1. h(Userpage)= A userpage containing options to manage music is shown.
2. h(Usermusicaddpage)= A music page is displayed, with choices to add music from various sources.
3. h(Musicformpage)= A form for adding music is provided, based on the user's preferred method of adding music.
4. h(MymusicPage)= A page containing music information organized as per artists and albums is displayed.
5. h(Nowplayingpage)= A page with current playing music is displayed.

Last.fm integration

X (states):

1. MyaccountPage
2. lastFmPage

X_0 (initial states):

1. MyaccountPage

U (actions):

1. Connect_lastfm
2. Disconnect_lastfm

F (transition relation):

State	Action	Nextsate	Obseravtion
MyaccountPage	Connect_lastfm	lastFmPage	A from to connect to lastfm is displayed
MyaccountPage	Disconnect_lastfm	lastFmPage	An option to disconnect from lastfm is shown

Y(observables):

1. A from to connect to lastfm is displayed
2. An option to disconnect from lastfm is shown

h (displaymap):

1. h(lastFmPage)=A page to connect and disconnect from lastFm is shown

Administrator features

X (states):

1. Adminloginpage
2. Adminpage
3. Directorypage
4. Transcoderpage
5. Sanitycheckpage
6. logspage

X_0 (initial states):

1. Adminloginpage

U (actions):

1. login
2. View_directory
3. Add_directory
4. Rescan directories
5. Delete_directory
6. View_transcoder
7. Add_transcoder
8. Delete_transcoder
9. Do_sanity check
10. View_logs

F (transition relation):

State	Action	Nextsate	Obseravtion
Adminloginpage	login	Adminpage	Admin homepage is displayed upon login
Adminpage	View_directory	Directorypage	Directed to directory page to display directories
Adminpage	Add_directory	Directorypage	Option to add new directory is shown
Adminpage	Rescan directories	Directorypage	Provides a page containing option to de re-scan
Adminpage	Delete_directory	Directorypage	Completes the delete operation of directory
Adminpage	View_transcoder	Transcoderpage	Directed to a page containing transcoder
Adminpage	Add_transcoder	Transcoderpage	A model to add new transcoder is shown
Adminpage	Delete_transcoder	Transcoderpage	A model to delete already existing transcoder is shown
Adminpage	Do_sanity check	Sanitycheckpage	Displays the page containing option to do sanity check
Adminpage	View_logs	logspage	Displays the page containing all logs

Y(observables):

1. Admin homepage is displayed upon login
2. Directed to directory page to display directories
3. Option to add new directory is shown
4. Provides a page containing option to de re-scan
5. Completes the delete operation of directory
6. Directed to a page containing transcoder

7. A model to add new transcoder is shown
8. A model to delete already existing transcoder is shown
9. Displays the page containing option to do sanity check
10. Displays the page containing all logs

h (displaymap):

1. h(Adminpage)= A page containing options to manage things is shown.
2. h(Directorypage)= A page with options to view, delete, rescan, and add directories is displayed.
3. h(Transcoderpage)= A page with options to view, delete, and add transcoder is displayed.
4. h(Sanitycheckpage)= A page with sanity check functionality is displayed.
5. h(logspage)=A page containing logs of the application is shown