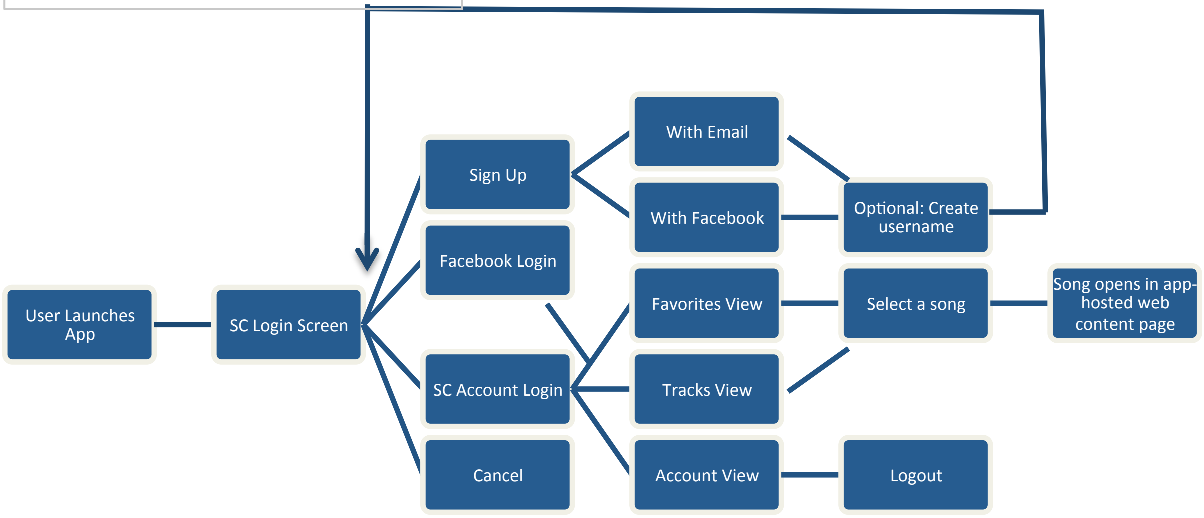
SoundCloud iOS Testing Challenge

User Scenario Flow Chart



Test approach:

1. Exploratory testing: manual testing aimed at testing all app features
   - Helps understand features and user scenarios
2. Unit Testing: application tests to test classes, functions, and interaction between classes
   - Test Automation: Unit Tests configured to run in Debug/Test mode to verify any code changes
3. Scenario Model-based Testing: manual testing aimed at testing user scenarios
   - Helps understand failure cases at each user state and how the user would be impacted
   - Helps prioritize critical (common) scenarios
4. User Experience testing: understanding different users and evaluating app usability
5. Other test considerations (Tenets)

For this challenge, I spent time studying the code architecture and user scenarios. I began testing with exploratory testing to survey the high-level app components.

I organized my unit test cases by thinking about the app as three parts: model, view, and controller. The following pages contain architecture diagrams and application unit tests for each part. I also intend to include the application unit tests in the application, so that any changes to the code can be validated automatically through a high-coverage automation test suite. Even without the automated test suite, the application unit tests can be run manually and verified through NSLog print statements.

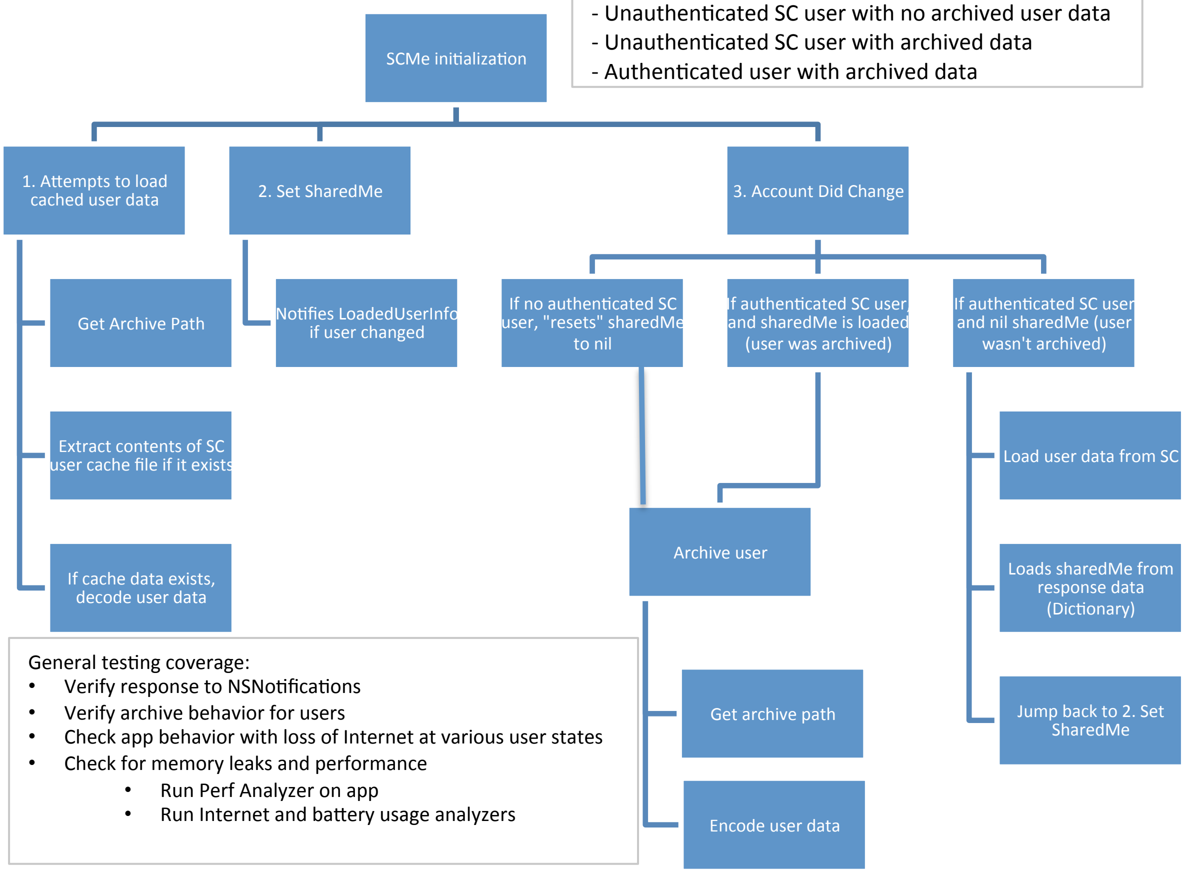
In addition to the unit tests, as I work as a user experience at Microsoft, I spent time evaluating the user experience of the app by looking at critical user scenarios (i.e. logging in and looking at favorites and tracks) and thinking about potential failure cases that could occur at each state in the above user scenario flow chart and how impactful the failure would be to the user.

Finally, I evaluated the app based on other test considerations, described later in detail.
As I am still relatively new to iOS development, I had a lot of fun and learned a lot working on this challenge! I plan to implement the unit test automation to further my understanding of the iOS testing tools. Please feel free to contact me with any questions at <Samantha.Luber@gmail.com>!

Code Architecture Maps

I. Model Architecture Map

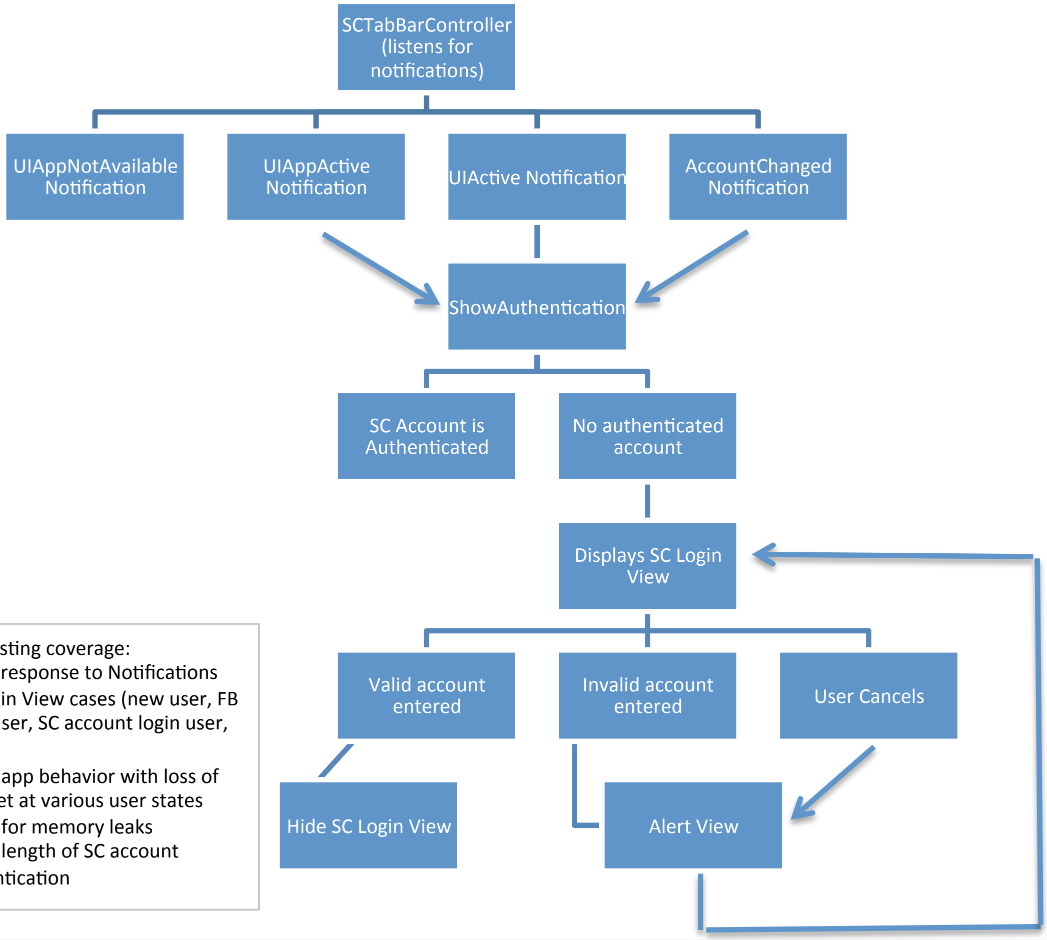
States of user at initialization:
- Unauthenticated SC user with no archived user data
- Unauthenticated SC user with archived data
- Authenticated user with archived data


SCMe initialization
1. Attempts to load cached user data
2. Set SharedMe
3. Account Did Change
Get Archive Path
Notifies LoadedUserInfo if user changed
If no authenticated SC user, "resets" sharedMe to nil
If authenticated SC user, and sharedMe is loaded (user was archived)
If authenticated SC user and nil sharedMe (user wasn't archived)
Extract contents of SC user cache file if it exists
Load user data from SC
Archive user
If cache data exists, decode user data
Loads sharedMe from response data (Dictionary)
Get archive path
Jump back to 2. Set SharedMe
Encode user data

General testing coverage:
- Verify response to NSNotifications
- Verify archive behavior for users
- Check app behavior with loss of Internet at various user states
- Check for memory leaks and performance
  - Run Perf Analyzer on app
  - Run Internet and battery usage analyzers

Model classes test cases:
- Case 1: No authenticated user and no archived user
  - Verify no cache file found in dearchive function
  - Verify no LoadedUserInfo notification sent from setSharedMe (sharedMe should be nil)
  - Login with a SoundCloud account (alternative: login with a Facebook account)
  - Verify AccountDidChange notification (state: sharedMe should still be nil, but SoundCloud account should be authenticated)
  - Verify user data loaded from SoundCloud servers (NSDictionary format)
  - Verify LoadedUserInfo notification sent from setSharedMe
  - Verify user info archived properly
- Case 2: No authenticated user and archived user (same user)
  - Verify cache file found in dearchive function
  - Verify sharedMe decoded from cache file
  - Verify LoadedUserInfo notification sent from SetSharedMe function
  - Login with same account as archived user (alternative login with Facebook)
  - Verify AccountDidChange notification
  - Verify user data is not loaded from SoundCloud
  - Verify user data is archived
- Case 2b: No authenticated user and archived user (different user)
  - Repeat steps 1-3 from case 2
  - Login with a different account than the cached user
  - Verify AccountDidChange notification
  - Verify sharedMe is set to nil and archive is cleared
  - Verify steps 4-6 from case 1

II. Controller Architecture



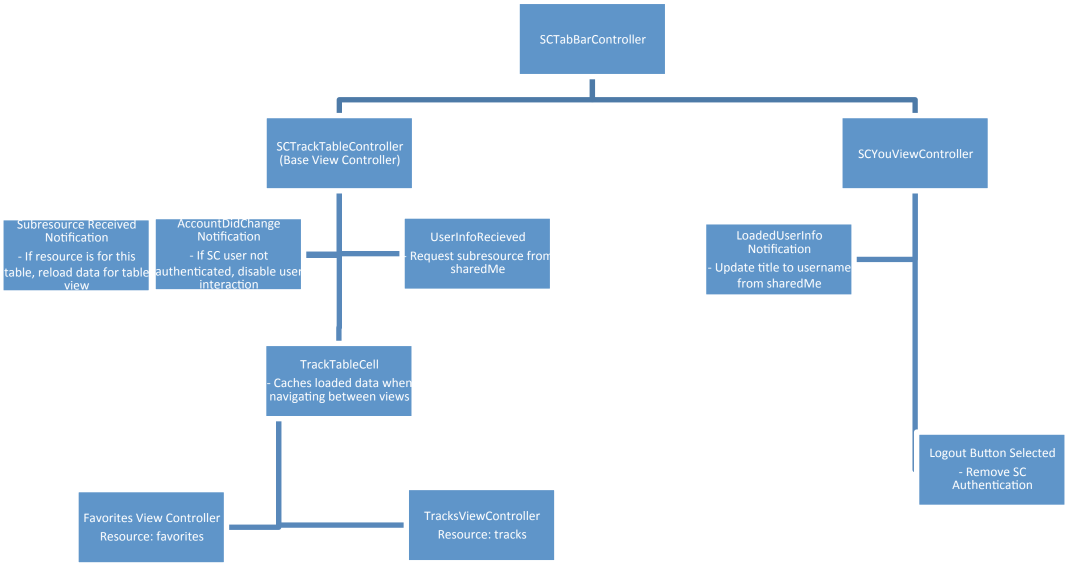
General testing coverage:

- Verify response to Notifications
- SC Login View cases (new user, FB login user, SC account login user, etc.)
- Check app behavior with loss of Internet at various user states
- Check for memory leaks
- Check length of SC account authentication

Controller classes test cases:

- Case 1: Notification Response
    - Launch App
    - Verify UIActive Notification called
    - Verify Login Screen shows for a non-authenticated SoundCLoud account
    - Enter invalid login information
    - Verify alert
    - Hit cancel
    - Verify alert
    - Enter valid login information
    - Verify Hide Authentication
    - Rotate device
    - Verify notification/response
    - Change SoundCloud account from an external source
    - Verify AccountChanged notification
    - Let app sleep
    - Open app
    - Verify UIAppActive notification
- Case 2: View Navigation
    - Verify Views Load
    - Verify subresources load
    - Verify avatar with account change from an external source
    - Verify username with account change from an external source

III. Controller/View Architecture


SCTabBarController
SCTrackTableController (Base View Controller)
SCYouViewController
Subresource Received Notification
- If resource is for this table, reload data for table view
AccountDidChange Notification
- If SC user not authenticated, disable user interaction
UserInfoRecieved
- Request subresource from sharedMe
LoadedUserInfo Notification
- Update title to username from sharedMe
TrackTableCell
- Caches loaded data when navigating between views
Logout Button Selected
- Remove SC Authentication
Favorites View Controller
Resource: favorites
TracksViewController
Resource: tracks

General testing coverage:

- Verify switching between views
- Verify response to Notifications handled by the correct controller in various views
- Verify device rotation
- Border tests (no tracks to display, max number of tracks to display)

Controller/view classes test cases:

- Case 1: SCYouViewController
    - Navigate to You page
    - Verify content
    - Change account username from an external source
    - Verify LoadedUserInfo notification
    - Verify You page content is updated
    - Logout
    - Verify account is un-authenticated
- Case 2: SCTrackTableController - Favorites
    - Verify content loads
    - Change favorites list on an external source
    - Verify UserInfoRecieved notification
    - Verify subresource request sent
    - Verify subresource received for this table
    - Verify reload data
    - Repeat steps 2-5 with a subresource received notification for a different resource (tracks)
    - Verify no reload data
    - Select a song
    - Verify that the app navigates to the in-app hosted SoundCloud display of the song
    - Navigate back to the Favorites view
    - Change account information to invalidate authentication
    - Verify AccountDidChange notification
    - Verify view becomes inactive
    - Stress testing (0 songs, MAX_SONGS, no internet connection)
- Case 3: SCTrackTableController - Tracks
    - Repeat Case 2 steps for Tracks view

Other test considerations:

- Accessibility: How does the app behave when the device has an accessibility setting on?
    - Ex: VoiceOver, Zoom, Large Text, Invert Colors, Speak Selection, Speak Auto-text, Hearing Aid Mode, Assistive Touch
- Globalization: How does the app behave when run on an iOS device in another language?
    - Is the font choice appropriate for other languages?
    - Does text fit into the control boxes properly?
    - In right-to-left read languages, do the control invert properly?
- Security
    - When is the user logged out of the app? When does the app re-authenticate?
    - For login with Facebook, SoundCloud hosts Facebook mobile site
        - Breaks user experience when attempting to navigate away from Facebook login page that is hosted inside SoundCloud iOS app
        - Security concern if Facebook or other hosted website downloads malicious data to device or stores user's information
- Performance
    - How long does the app take to load?
    - How long does it take to switch between views in the app?
    - How is app performance affected when many other iOS applications are running in the background?
    - How is app performance affected by a slow Internet connection?
    - How much memory does the app use on the device?
    - How much CPU processing power is consumed by the app?
    - How often does the app reload its data from the SoundCloud servers?
    - Does the app cache user information? If so, how much and for how long?
    - Does the app dramatically affect battery life of the device?
- Compatibility
    - How does the app scale on iOS devices of different sizes? (i.e. iTouch, iPhone 3, iPhone 4, iPhone 4S, iPhone 5, iPad, iPad retina, iPad mini, etc.)
- Reliability
    - What does the app experience look like with no Internet connection?
    - What happens if Internet connection is lost while the user is using the app?
    - What happens if the user updates their account or changes their favorites list on an external device? Does the iOS app update its information?
    - What happens if the artist deletes a song I have favorited while I am using the app? Does the song disappear? If not, what happens when I click on the song to navigate to it in a web browser?
    - Does the app store user state? If I navigate away from the app and reopen it, am I taken back to my current view?
    - What happens if I delete my SoundCloud account from an external source while using the app as a logged in user?
    - What does the user experience look like if the SoundCloud servers are down?
- User Experience
    - When logging in with Facebook, no way to navigate back if I click to view Facebook's privacy policy
    - In Forgot Password view, "Oh I remember it" is less intuitive to navigate back to the main page than a simple "Cancel" button
    - App fails when user attempts to navigate to another webpage from hosted Facebook mobile page in SoundCloud app
    - Difficult to read song names in favorites list
    - Favorites icons could be improved
        - Musical note instead of a circle for Tracks?
        - Silhouette instead of a square for user?
    - Persistent Cancel button in Login Screens confusing
        - Two cancel buttons on Login page
        - Cancels and leaves app from any view in Login page (not what the user expects?)
    - Zoom on Terms of Use and Privacy Policy screens breaks user experience

Issues found:

- Fails when attempting to sign up for SoundCloud using Facebook account
    - Repro steps: Launch app, click Sign Up tab, click "With Facebook" button
- SoundCloud app attempts to host web browser content outside of Facebook mobile site
    - Repro steps: Launch app, under Sign in click "With Facebook", click on "Soundcloud" link at the top or "Facebook app" link below
- In iPhone retina hardware simulator, "With Facebook" login button doesn't scale (looks stretched/ pixely)
- App isn't compatible with iPad/iPad retina