Sprint 2 Retrospective - Iguana Team

Team 1

Alec Gorge, Manik Kalra, Scott Opell, Andrew Shildmyer, Kirby Kohlmorgen

1. **What went well**
    a. The web app talks to the server API's to fetch artists, years, venues, and songs.
    b. The web app now let's you share songs on Facebook.
    c. The Mac App has a reorderable queue for managing and playing music.
    d. The server has API's for playlists on the web app, Mac app, and iOS app.
    e. The iOS app has an elegant interface and support for viewing top shows and random shows.
2. **What did not go well**
    a. Time management/coordination. For example, the final API routes were not working until hours before the deadline, which blocked the iOS and web apps from getting certain things implemented.
    b. The was an underestimation on how long it would take to create the UI for song selection in the web app, that ultimately prevented us from implementing playlists in the web app.
3. **How we should improve**
    a. Communicate better and set internal deadlines on certain items.
    b. Put more time on our estimates, things always take longer than you think.

1. **User story: As a user, I would like to sign into and register for a relisten.net account in the iOS app.**

Task Description	Time	Owner	Completed
Fix iOS UI bugs	7	Manik	Yes
Add Login/Signup UIs	8	Manik	Yes
Consume API and create logic for user accounts	10	Manik	Almost. Did not have have the API routes on time
Add support for top-rated and random shows for multiple artists	5	Manik	Yes
Test API rigorously and under load	8	Scott	Yes

Change API models to support authenticated API methods	5	Alec	Yes
Change API clients to support authenticated API methods	6	Alec	Yes
Securely store and retrieve username, password and other user information	4	Alec	Yes

2. **As a user, I would like to have gapless playback in the iOS app.**
    a. Completed
3. **User story: As a user, I would like to access all the artists in one web app.**
    a. Completed
4. **User story: As a user, I would like to have playlist support in the web app.**

Task Description	**Time**	**Owner**	**Completed**
Add playlist UI to the add.	6	Andrew	No
Add javascript to the UI for reordering playlist items.	5	Andrew	No
Add ajax to dynamically display Playlist items.	7	Kirby	No
Add the ability to sort by fields.	8	Kirby	No
Integrate playlist API into the backend application	6	Scott	Yes

5. **User story: As a user, I would like to have the ability to share in the web app.**
    a. Completed
7. **As a user: As a user, I would like to have an easy-to-use, user-friendly web interface.**
    a. Completed
8. **As a user, I would like to be able to manage playback of music in a reorderable queue in the OS X app.**

Task Description	**Time**	**Owner**	**Completed**
Build basic queue UI	4	Alec	Yes
Allow queue rows to be reordered by dragging and dropping	4	Alec	No
Add button to remove from the queue	1	Alec	No
Add button to open track in show context	3	Alec	No

Add buttons to track playback that allows for adding to the beginning and end of the queue as well as replacing it	4	Alec	Not accessible via UI

9. **Maintenance / Other tasks**
    a. Completed

From Sprint 1:

1. **User story: As a user, I would like to have a playlist in the iOS app.**

Task Description	**Time**	**Owner**	**Completed**
Design database and API to handle playlists	5	Scott	Yes
Implement API methods to create, update and delete playlists	7	Scott	Yes
Add playlist support on the iOS app	7	Manik	Part of Sprint 3
Do code cleanup for more reusability and clarity, so that adding new features is easy	2	Manik	Yes