

**Team 18 - Sprint 1 Planning Document**

Andrew Furman, Nick Galecki, Kevin Matson, Josh Shepler, Doug York

**Sprint Overview**

During this sprint, our primary objective is to implement basic backend functionality on the server side of our program, as well as create a basic webapp client. The backend will consist of a server program as well as a SQL database, which will handle logic and data storage functionality. The client will consist of GUI screens similar to those in the design document, as well as code facilitating communication with the backend server. We will need to set up classes for communication with each of the streaming services we hope to support, and then link them to the backend. By the end of this sprint we will have a desktop client which will allow users to connect to the server to create and access InterLinked accounts. The server will also contain methods for connecting to a users Spotify account. They will also be able to import and export playlists from Spotify.

**SCRUM Master**: Doug York

**Meeting Plan:** Mondays 6:00PM-7:00PM, Thursdays 3:30PM-4:30PM, Weekends as needed.

**Risks and Challenges**

Challenges in this sprint will mostly include difficulties in connectivity between the GUI, Server and Database. Ensuring that these three things are connected successfully will require that the majority of our unit tests function correctly. Our unit tests will help verify that our implementation of various features are functioning as intended. Another challenge with this sprint will be communication amongst team members. We will have to have strong communication as a team especially when combining parts of the project done by different individuals. A risk we could run into during this sprint could be failing to develop all the user stories included in this planning document. Our team doesn’t have a lot of experience with using databases, which could lead to a lot of time being spent figuring out how to connect all of the aspects of our project together using a database. Another risk we could run into would be having transfer rates limited by Spotify when importing, exporting, and merging playlists. We are not familiar with using the Spotify API and do not know how much Spotify throttles those who are developing with their API.

**Current Sprint Detail**

User Story 1:

As a user, I would like to sign up for InterLinked.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Implement a Login/Register GUI screen | 2 | Josh, Kevin |
| 2 | Create a User class | 1 | Andrew |
| 3 | Setup a database using SQL Server | 3 | Nick |
| 4 | Add a User table to the database | 1 | Nick |
| 5 | Read username and password from the GUI. Store them as a user object in the User table on the database using salted hashes | 3 | Nick |
| 6 | Unit tests- Create an account and make sure there are notifications for when the account creation/login process is unsuccessful, also check that the users information is saved in the database | 2 | Nick |

Acceptance Criteria:

* Given that the user login/register functionality is implemented correctly, a user can obtain access to the login/register page of InterLinked.
* Given that the database is setup correctly, a new users information can be entered into the User table in the database.
* Given that a user creation request is sent, then the server should parse the request and check it for correctness, finally pushing the results to the database.
* Given that a user account is unavailable (already existing or some other precondition), then the user will be notified in the GUI.
* Given that a user account creation process successfully, then the user will be notified in the GUI.

User Story 2:

As a user, I want to save my playlists.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Create Playlists and Songs tables in the database | 2 | Nick |
| 2 | Implement Playlist and Song classes on the server | 2 | Nick |
| 3 | Implement a method to create a Playlist object, and a method to add Song objects to the Playlist | 2 | Andrew |
| 4 | Store the Playlist object onto the database in the Playlists table | 1 | Nick |
| 5 | Create Playlist and Song classes on the web app | 2 | Josh, Kevin |
| 6 | Obtain playlists from the server | 2 | Nick |
| 7 | View saved playlists in the GUI | 3 | Josh, Kevin |
| 8 | Unit Test - Save a playlist server-side and through the GUI. Ensure that it is actually saved on the database | 2 | Nick |

Acceptance Criteria:

* Given a playlist on the user’s account, when the user logs in to their account, then they would see the stored playlists.
* Given the database has playlists in it from various users, when the user logs in to their account, other users’ playlists are not visible.
* Given a user with no playlists, when the user logs in, the GUI playlist list will be empty in a way that is clear to the user.
* Given a user has multiple playlists, the user can see all of their playlists and view any one of them at a time.

User Story 3:

As a user, I want to search for songs from within the app.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Create a search GUI. Select which streaming service should be searched on the GUI | 3 | Josh, Kevin |
| 2 | Format the query within the specialized Streaming Service class | 3 | Doug |
| 3 | Send the query to the selected service and display the result | 5 | Doug, Josh, Kevin |
| 4 | Send the results from the server to the client | 2 | Andrew |
| 5 | Display the results on the client | 4 | Josh, Kevin |
| 6 | Unit test - The search produces results that can be found on the designated supported platforms | 2 | Josh, Kevin |

Acceptance Criteria:

* Given a search, relevant search results from the supported platforms will be displayed.
* Given a search, where the user chooses to search a particular platform, only results from that platform will be displayed.
* Given a search, where a result is not found, the user will be notified.
* Given a search, where the user instead writes an SQL query into the search bar, the SQL query will not be executed on the database of InterLinked (protection from SQL Injection).

User Story 4:

As a developer, I want an API that allows for universal importing from sites without needing a separate method for each site.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Create an abstraction class that has unimplemented (for now) methods we can use to interact with different platforms | 2 | Andrew |
| 2 | Connect to a User’s Spotify Account | 4 | Doug |
| 3 | Unit Test - The Stream Service class hides the details of implementation of the Spotify instance | 1 | Andrew |

Acceptance Criteria:

* Given the Spotify class inherits the StreamingService class, the programmer can use its methods without worrying about the mechanics of the Spotify API.
* Given a change is made in the Spotify class, it is only to fix bugs or to extend the implementation, while keeping the base results the same.
* Given a method that’s been inherited from the StreamingService class is called, it behaves the same way regardless of what service the method is implemented for.

User Story 5:

As a user, I want to import a playlist from Spotify.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Implement the import method in the abstract streaming service class for Spotify | 2 | Doug |
| 2 | Copy the list of songs, and parse the relevant information into Song objects | 6 | Andrew |
| 3 | Create a playlist object, then save it in the database | 2 | Andrew |
| 4 | Unit test - Import a playlist from Spotify using our method and verify that the parsing we used is correct | 3 | Andrew, Doug |

Acceptance Criteria:

* Given that a user has not connected their Spotify account, when they select the option to import a playlist from Spotify, then they will be prompted by the GUI to login first.
* Given that a user has connected their Spotify account, when they select the import playlist option, a list of their playlists on Spotify will be displayed on the GUI.
* Given that a user successfully imports a playlist, then their playlist will be added to their saved playlists in the database.
* Given that the playlist is imported successfully, then the user will be notified that the playlist has been added to their collection of saved playlists.

User Story 6:

As a user, I want to export a playlist to Spotify.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Find out how the Spotify API recognizes songs | 4 | Doug |
| 2 | Create the Spotify version of the export playlist method in the abstract streaming service class | 4 | Doug |
| 3 | Unit test - Use the method to export a playlist object to Spotify and verify that it works correctly | 2 | Doug |

Acceptance Criteria:

* Given that a user has at least one playlist, when they select the export playlist option, a list of their playlists will be displayed for export.
* Given that a user has exported a playlist, when they view the playlist on that platform, it will contain the correct songs.
* Given that a user is exporting a playlist, when a song cannot be found on the destination platform, then the user will be notified through the GUI that their playlist will be incomplete.

User Story 7:

As a user, I want to delete a playlist.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Display the user's stored playlists, and select one for deletion | 3 | Josh, Kevin |
| 2 | Remove the playlist entry from the Playlist table | 2 | Nick |
| 3 | Update the list of playlists on the GUI | 3 | Josh, Kevin |
| 4 | Display a confirmation message on the GUI if deletion was successful | 2 | Josh, Kevin |
| 5 | Unit test - Delete a saved playlist and make sure it is removed from the user’s list of saved playlists in the database | 2 | Josh, Kevin |

Acceptance Criteria:

* Given that a user wants to delete a playlist, they will be able to select which playlist they want to remove.
* Given there was a successful deletion, the deleted playlist should no longer be in the database.
* Given there was a successful deletion, when the database is queried, the playlist will not be found.
* Given there was a successful deletion, the user should no longer see the deleted playlist in their saved playlists.
* Given there was a successful deletion, the user will be notified that their selected playlist was deleted.

User Story 8:

As a user, I want to merge two or more playlists into a new playlist.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Make a method that merges two playlist class objects into one object | 3 | Andrew |
| 2 | Save the generated playlist object | 1 | Andrew |
| 3 | Display confirmation UI screen that shows outcome of merge request | 2 | Josh, Kevin |
| 4 | Unit tests - merge two playlists and verify that our method works correctly under normal conditions and corner cases | 2 | Josh, Kevin |

Acceptance Criteria:

* Given the user has at least two playlists, when they select the merge playlists option, a list of their playlists will be displayed for merging.
* Given that two playlists are merged successfully, when the new playlist is viewed, it will contain all songs in both of the source playlists.
* Given the user has less than two playlists on their account, the merge playlists option will be unavailable in a way that is clear to the user.
* Given the merging playlists have the same song on in, when the user merges the playlists, only one instance of the song is present.

User Story 9:

As a user, I want to open a searched song on the corresponding platform for listening.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Display a GUI for a user to enter information about the song to be searched. | 2 | Josh, Kevin |
| 2 | Format and send a query to the Spotify API | 2 | Doug |
| 3 | Use the returned link to open the song in the Spotify app. | 2 | Doug |
| 4 | Testing - Open different songs using the Spotify API to make sure they work correctly | 2 | Josh, Kevin |

Acceptance Criteria:

* Given that the user has searched for a song that is available on Spotify, that song will be played, either in the browser or app depending on that user’s configuration.
* Given that the user has searched for a song that is not available on Spotify, the option to play a song will not be available for a clear reason.
* Given that the user listens to a song in Spotify, when they switch apps, then they will not be logged out of InterLinked.

User Story 10:

As a user, I want to add a searched song to one of my playlists.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Display the results of a search query for the user to select a song. | 2 | Josh, Kevin |
| 2 | Display the user’s current playlists that the song can be added to. | 2 | Josh, Kevin |
| 3 | Add the song to the selected playlist in the playlists table. | 2 | Andrew |
| 4 | Testing: Search for and add several songs to playlists, ensuring the database is updated accordingly. | 2 | Nick |

Acceptance Criteria:

* Given a user has searched for a song, when a song is found, they will have the option to add that song to a playlist.
* Given a user has searched for a song, when they currently have no playlists, they will be able to start a new playlist with that song.
* Given a user has chosen to add a song to a playlist, when that playlist already contains that song, then the user will be notified of the duplicate and the song will not be added.

User Story 11:

As a user, I want to sort my playlists by different criteria.

Task Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Description | Time | Task Owner |
| 1 | Display a list of the user’s playlists on the GUI for them to choose from. | 2 | Josh, Kevin |
| 2 | Display options for which field to sort by (Artist name, Song title, Album title, Genre) | 2 | Josh, Kevin |
| 3 | Sort the playlist on the backend Server | 4 | Andrew |
| 4 | Display the sorted playlist to the user. | 2 | Josh, Kevin |
| 5 | Testing: Sort several different playlists using our implementation and ensure the order is correct. | 2 | Doug |

Acceptance Criteria:

* Given that a user has sorted a playlist by artist name, when the songs are displayed, then they will be ordered alphabetically by the artist name.
* Given that a user has sorted a playlist by song title, when the songs are displayed, then they will be ordered alphabetically by the song title.
* Given that a user has sorted a playlist by album title, when the songs are displayed, then they will be ordered alphabetically by the album title.
* Given that a user has sorted a playlist by genre, when the songs are displayed, then they will be organized by musical genre.

**Remaining Product Backlog**

1. As a user, I want to know which music streaming services are supported.
2. As a user, I want to import a playlist from Amazon Music.
3. As a user, I want to export a playlist to Amazon Music.
4. As a user, I want to import a playlist from Apple Music/iTunes.
5. As a user, I want to export a playlist to Apple Music/iTunes
6. ~~As a user, I want to import a playlist from Spotify.~~
7. ~~As a user, I want to export a playlist to Spotify.~~
8. As a user, I want to import a playlist from Youtube.
9. As a user, I want to export a playlist to Youtube.
10. As a user, I want to be notified when a playlist is imported successfully.
11. As a user, I want the transfer of playlists to be continued if it is interrupted.
12. As a user, I want to be notified when a playlist is exported successfully.
13. As a user, I want to be notified when a song can't be exported to a site.
14. ~~As a user, I want to search for songs in the app.~~
15. ~~As a user, I want to open a searched song on the corresponding platform for listening.~~
16. As a user, I want to choose which platforms I search from.
17. As a user, I want to filter search results.
18. As a user, I want to add songs from Amazon Music to my playlist.
19. As a user, I want to add songs from Apple Music/iTunes to my playlist.
20. ~~As a user, I want to add songs from Spotify to my playlist.~~
21. As a user, I want to add songs from Youtube to my playlist.
22. ~~As a user, I want to remove songs from my playlist.~~
23. ~~As a user, I want to save my playlists.~~
24. ~~As a user, I want to delete my playlists.~~
25. As a user, I want to revert a playlist back to a state it was in the past.
26. ~~As a user, I want to organize my playlists by song title.~~
27. ~~As a user, I want to organize my playlists by album title.~~
28. ~~As a user, I want to organize my playlists by artist name.~~
29. ~~As a user, I want to organize my playlists by music genre.~~
30. ~~As a user, I want to merge two or more playlists into a new playlist.~~
31. As a user, I want the app to listen to a song and put it into my playlist.
32. As a user, I want to share my playlist with a friend.
33. As a user, I want to suggest songs to a friend.
34. As a user, I want the program to suggest songs to add to my playlists.
35. ~~As a user, I want the program to run on a computer.~~
36. As a user, I want the program to run on an Android device.
37. As a user, I want the app playlists to be transferable cross-platform.
38. ~~As a user, I want the GUI to be intuitive and simple.~~
39. As a user, I want to have my playlists automatically synchronized between platforms.
40. ~~As a user, I want to sign up for InterLinked~~
41. As a developer, I want to ensure security of user data.
42. ~~As a developer, I want an API that allows for universal importing from sites without needing a separate method for each site.~~
43. ~~As a developer, I want an API that allows for universal exporting to sites without needing a separate method for each site.~~
44. As a developer, I want the playlists to be independent from their source(s).