

## Iteration 23

1. Develop as much as possible the **functionality** described in chosen **Product Backlog**
2. Zip file named 'Iteration2and3\_XX' containing
3. Revised **Product Backlog** that clearly label **which** of the **stories / tasks** you have **finished** in this iteration. The **revised Product Backlog** should clearly indicate what was the **estimated** and what was the **actual effort** for each of the **finished stories / \*-tasks**. A diagram showing your **team velocity over time** is mandatory
4. A **test plan** indicating **tests** with **test conditions**, **test inputs**, and **expected outputs**
5. A **test report** indicated which **tests** were **run** and **which** of them have **passed**
6. Complete **source code** of your project, with **all** the **files** needed
7. 'ReadMe.txt' file that describes the delivery package, including the names of team members who have contributed to the iterations

## Product Backlog

Task	Estimate	Implementation Notes	Classes
<b>Linked list</b> contain <b>desired songs</b> for a <b>playlist</b> and will be used to <b>save the playlist</b> . <b>Read into a queue</b> to be <b>played</b>	2 hours		playlist
<b>Song queue</b> will have <b>limitations</b> that require the <b>total time of songs</b> to be <b>between 1 to 3 hours</b> . Otherwise the <b>queue will not be able to be playable</b>	30 minutes		playlist
<b>User</b> classified as the <b>administrator</b> will be able to <b>select a button</b> on the <b>song library</b> that will <b>allow the admin</b> to <b>select songs</b> for <b>restricted access</b>	1 hour		User
<b>Drop down menu</b> implemented for any <b>classification necessary</b> , with <b>list of options</b> . <b>Selection options</b> should accurately <b>search the list</b> and <b>find songs</b> which are of <b>those properties</b>	3 hours	1 hour	GUI
A <b>toolbar</b> implemented which <b>contains the menus</b> and <b>allows</b> for	1.5 hours		GUI

<b>expansion for future classification menus</b>			
Tool will do <b>smart search</b> which take <b>users previous selections of songs</b> , <b>dissect its traits</b> and <b>compare</b> them to <b>other songs</b> in the <b>database</b> , actively <b>displaying the results to the user</b>	3 hours		GUI Playlist
<b>Randomizing tool</b> will be <b>available</b> somewhere <b>near the top of the song browser GUI</b> , with <b>check indicating it is active</b>	1 hour		GUI
<b>While tool is active</b> , set of <b>classifications will be available</b> (genre, musician, mood) and are <b>selectable</b>	2 hours	Task complete within the set time of 2 hours.	GUI Playlist
<b>Queue auto fills with randomly selected songs, without repeat</b> , if possible. The <b>songs must follow the restriction set by users</b>	2.5 hours		Playlist User
Next to <b>every song</b> , a <b>statistics page button</b> will be <b>available to bring them to a page with that song stats</b>	2 hours		Stats GUI
<b>Relative frequency stats</b> section will be <b>available</b> and it will have a <b>time frame</b> selection by <b>utilizing a Google calendar</b> . The songs <b>relative frequency</b> in that time period will be <b>displayed as a line graph</b> with the <b>y-axis</b> being <b>relative frequency in percentage</b> and the <b>x-axis</b> representing <b>time</b>	4 hours		Stats
<b>A list will be created for every user</b> , with <b>basic information of each song that they played</b>	2 hours		User
A <b>time frame</b> can be <b>selected</b> from a <b>drop down menu containing time</b>	1.5 hours		GUI

frames, such as <b>1 day, 1 week, all time</b> , etc			
A <b>statistics page</b> will be <b>automatically</b> created for each <b>user</b> account <b>created</b>	2 hours		Stats User
<b>Pages of stats</b> will be <b>available</b> on the <b>user's stats page</b> , with a <b>tool bar</b> located to the <b>left</b> to <b>select</b> the <b>desired page</b> . This <b>side bar</b> must be <b>customizable</b>	1.5 hours		Stats User GUI
Since <b>saved playlists</b> utilize a <b>linked list</b> , <b>adding</b> , <b>removing</b> and <b>general reorganization</b> of <b>playlist</b> will be done using <b>typical list modifications algorithms</b> such as <b>changing required pointers</b> from <b>song to song</b>	1.5 hours		Playlist
<b>Currently active playlists</b> utilize a <b>queue</b> , so a <b>user</b> will have the <b>ability</b> to <b>designate</b> an <b>upcoming song</b> to be <b>skipped</b> with a <b>button</b> on the <b>GUI</b>	1 hour		GUI
The <b>queue</b> will <b>accept addition to the playlist</b> on the fly by <b>adding any new songs</b> to the <b>end of queue</b>	1 hour		GUI Playlist
Each <b>user</b> be given <b>hash table</b> upon <b>creation of an account</b> which will <b>store all user's playlists</b> . The <b>keys</b> for <b>each table</b> will be the <b>user's unique user name</b> in <b>conjunction</b> with a <b>unique playlist name</b> per playlist	5 hours		User
For <b>each current user</b> , the <b>head administrator</b> will be <b>able to access</b> a <b>"make admin"</b> option in the user's option pane	1.5 hours		Admin GUI

Test Conditions	Test Inputs	Expected Output
Signing up for an account	Clicking the signup button	Brings the user to the signup page
Unsuccessful signup because the passwords don't match	Password: 123456 Confirm Password: 12345	Produce an error message that says "Password does not match"
Successful signup	Username: newuser Password: abc Confirm Password: abc Press the signup button	Produce a message that says "Signup Successful" and brings user back to the login page
User leaves a field empty on the signup page	The username field is empty Password: abc Confirm Password: abc	Produce a message that says "Missing username or password"
User enters the correct username and password and selected the "User"	Username: newuser Password: abc Select the "User" option then press the login button	A message box pops up saying "Login Successful" and takes the user to a new page.
User enters the incorrect username	Username: newuse Password: abc	A message box pops up saying "Username does not exist"
User enters the incorrect password	Username: newuser Password: abcd	A message box pops up saying "Incorrect Password"
User enters correct username and password, but selects the wrong option	Username: newuser Password: abc Select the "Admin" option then press the login button	A message box pops up saying "You're not authorized"
Admin enters the correct username and password	Username: admin Password: 1234 Select the "Admin" option then press the login button	Takes you to the admin page
Admin enters the correct username and password, but selects the wrong option	Username: newuser Password: abc Select the "User" option then press the login button	Produce a message that says "Not a user account"
User/admin leaves a field empty on the login page	The username field is empty Password: 1234	Produce a message that says "Missing username or password"