Use Case Document

Introduction

This document contains a collection of use cases. It starts off with a high-levelled view of the system and is then followed by use cases that are divided into sections for each of the main components of the proposed Jammin music composition software. The use cases are written in a language that can be understood by target users who are people with a basic understanding of music composition and theory.

Index

1. General Use Cases

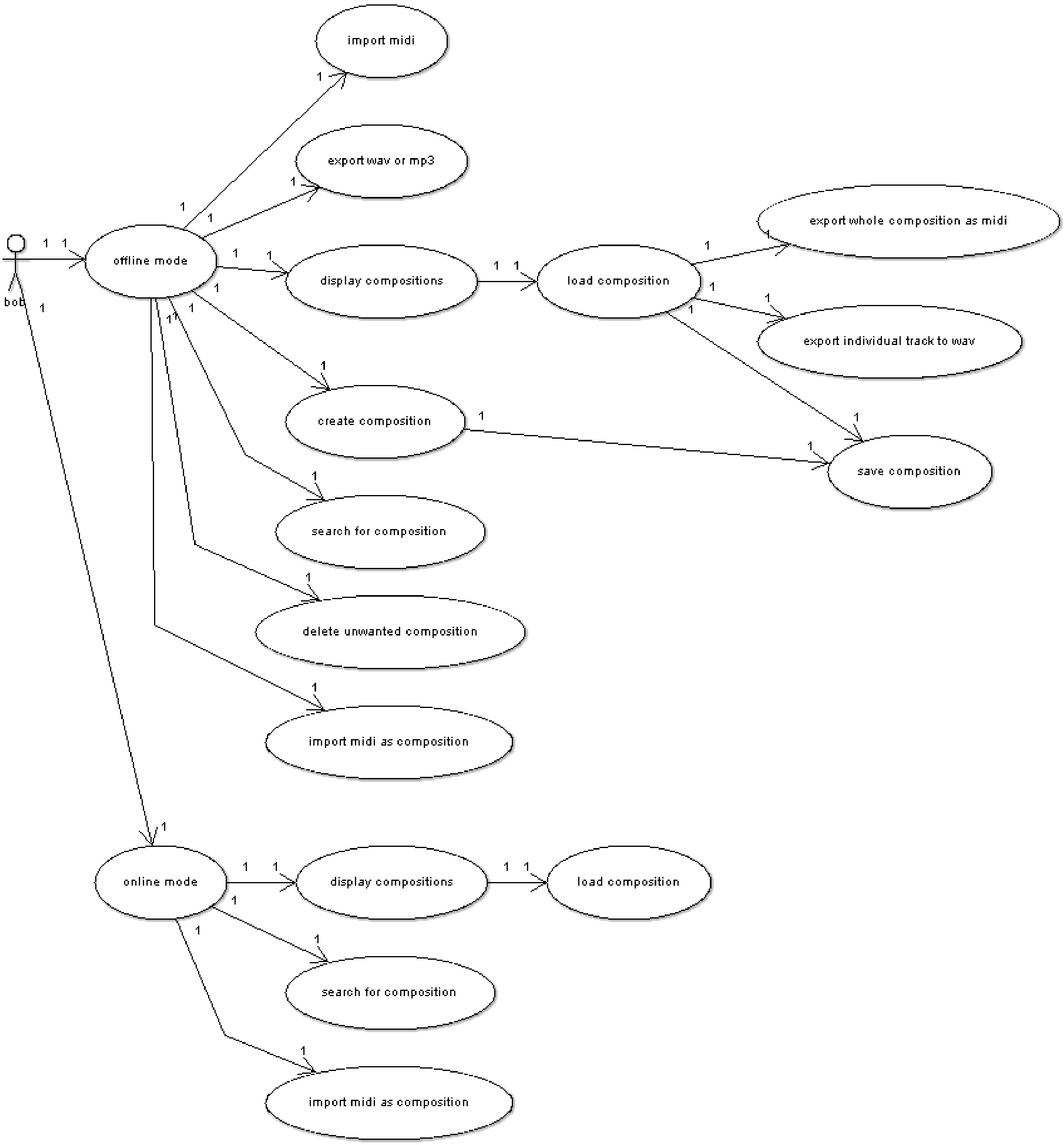
This section provides some high-level viewed use cases of the system. These use cases illustrate the core characteristics of our system.

Use Case	**Create a composition**
Description	A user wants to create a new composition
Actors	User-A – The person creating a new composition
Assumptions	User-A has opened the program
Steps	1. User-A creates/opens a blank composition. 2. User-A enters a melody into the composition. 3. User-A generates chord track to go with the melody.
Variations	**2b:** Creates/enters a chord track first then does variation 3b **3b:** Creates/enters a melody over the chord track.
Non-functional	none
Issues	none
Exceptions	none

Use Case	**Discuss and share a composition**
Description	A user wants to share a composition with a friend and discuss the composition (such as get their friend's suggestions on how to improve the composition)
Actors	User-A – The person who wants to share their composition with their friend (User A). User-B – The friend of User-A
Assumptions	User-A and User B are have opened the program and signed in online. User-A and User-B are on each others friends list.
Steps	1. User-A invites User-B to download their composition. 2. User-B accepts User-A's invitation. 3. User-A sends their composition to User-B 4. User-A starts a conversation with User-B and sends them a text message asking them for their thoughts of their composition. 5. User-B replies by sending a text message to User-A.
Variations	**2b:** User-B declines the invitation. The process stops.
Non-functional	none
Issues	none
Exceptions	• The sending of the file process fails because User-A or User-B goes offline. • User-A or User-B goes Offline during the conversation, then the conversation ends.

Use Case	**Assist beginner musician in practicing to play songs**
Description	A user wants to practice a song using a backing-track composition.
Actors	User-A – The person who wants to practice a song
Assumptions	User-A has opened the program
Steps	1. User-A opens the backing track that they want to practice with. 2. User-A gets the system to analyse a specific track and displays the chords for the track. 3. User-A mutes the selected track (in step 2). 4. User-A plays the composition and plays along with it (using the chords that the system discovered to aid them).
Variations	**2b:** User wants to learn the melody track so does not need to get the system to analyse and display the corresponding chords.
Non-functional	none
Issues	The selected track to analyse for chords is a melody track. The user is notified that the chords displayed are more suggestions rather than the actual chords that were intended to be played.
Exceptions	none

2. Managing Compositions



Use Case	**Create a blank composition**
Description	A user wants to create a new composition ASAP.
Actors	User-A – The person creating a new composition
Assumptions	User-A has opened the program and just logged in with their credentials.
Steps	1. User-A selects an action to create a new composition. They are not required to name their composition before creating it because of the artistic nature of music. 2. The new composition is displayed in the composition library if it is visible.
Variations	none
Non-functional	none
Issues	If User-A has an unsaved composition open, they will need to be prompted to save or discard the changes or cancel creation of a new composition and return to the precondition state. Alternatively the software may allow users to open and work on multiple compositions at once. The composition library will indicate which compositions are currently open.
Exceptions	It may be better to automatically open a blank composition already there every time User enters offline mode?

Use Case	**Open a Composition**
Description	A user wants to open a composition they previously worked on but they can't remember the name.
Actors	User-A – The person loading their composition
Assumptions	User-A has opened the program
Steps	1. User-A browses through composition library and actions to open an existing composition.
Variations	**1b**: The user can choose to search instead of browse.
Non-functional	none
Issues	None
Exceptions	none

Use Case	**Delete a Composition**
Description	A user wants to delete an old unwanted composition completely and permanently from disk.
Actors	User-A – The person deleting a composition
Assumptions	User-A has opened the program
Steps	1. User-A selects an existing composition. 2. User-A actions to delete the selected composition. 3. The system ask User-A to confirm whether or not they want to delete the selected composition. 4. User-A chooses to confirm 5. The file is removed from the composition library view.
Variations	**4b**: User-A chooses to cancel, so the process ends.
Non-functional	none
Issues	The file has already been removed from the hard disk.
Exceptions	

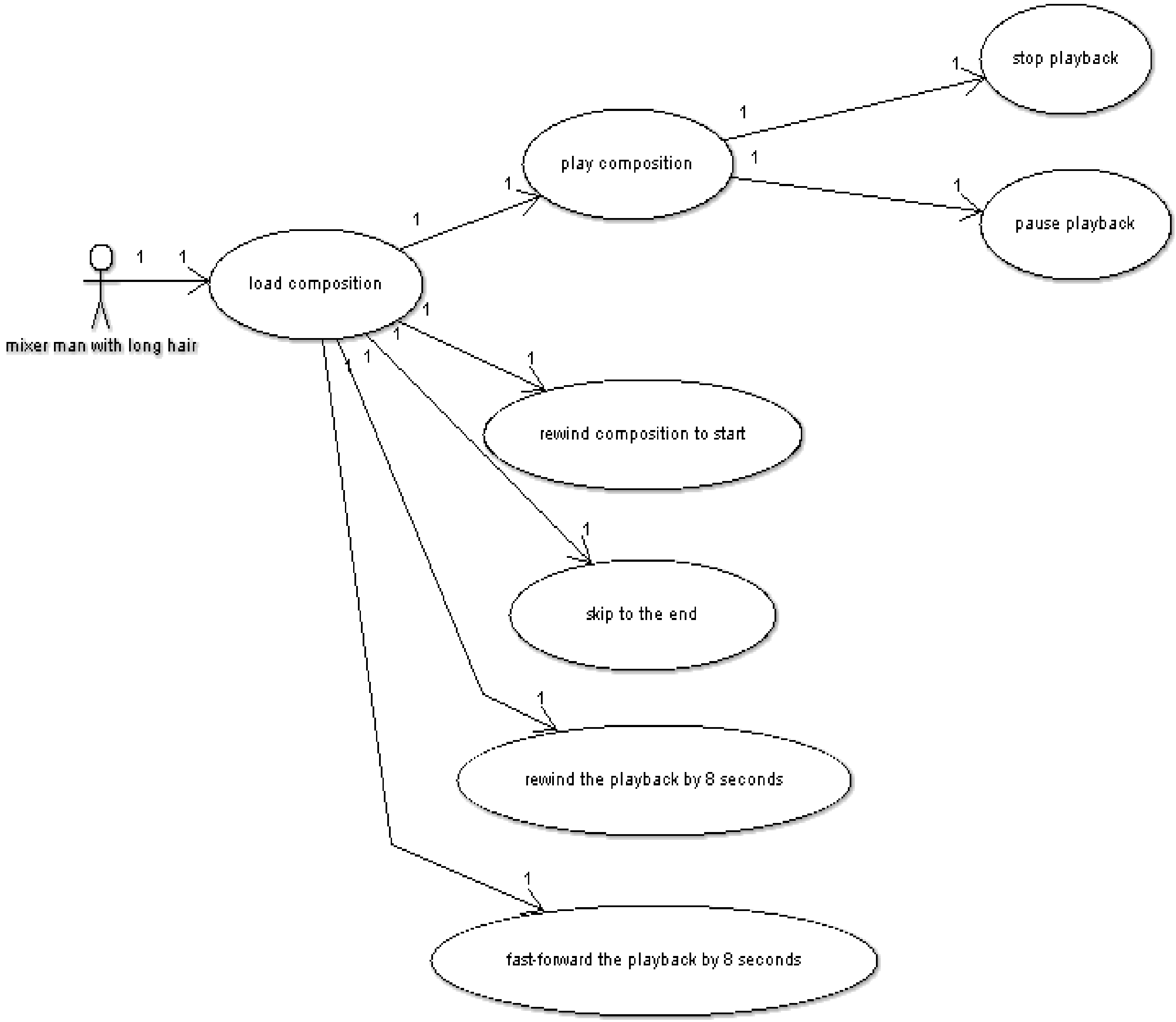
Use Case	**Save a Composition**
Description	A user wants to save a composition they are currently working on.
Actors	User-A – The person saving their composition
Assumptions	User-A has a composition open
Steps	1. User-A clicks the save button on the toolbar. 2. The composition is saved in place.
Variations	none
Non-functional	The save needs to be fast enough that the user can get back to work with very little delay.
Issues	none
Exceptions	If no name is given, the composition will need to be saved under a default name or ask the user for a name (invoke Save As...)

Use Case	**Search for a Composition**
Description	A user wants to search for a previously created or imported composition.
Actors	User-A – The person searching for a composition
Assumptions	User-A has opened the program The composition library is visible.
Steps	1. User-A inputs the name of the composition to search for. 2. User-A actions to search. 3. User-A browses the search results.
Variations	none
Non-functional	The search needs to be instant (less than 50ms), even if User has 4000 compositions in their library.
Issues	
Exceptions	There are no items that match the search term, in which case User should be informed of that result.

Use Case	**Export a Composition as a Midi File**
Description	A user wants to save their composition as a general midi file instead of the custom file format for the program.
Actors	User-A – The person exporting their composition
Assumptions	User-A has a composition made from only midi tracks.
Steps	1. User-A actions to export the actively opened composition. 2. A dialog appears asking them where on their computer they would like to export the midi file to.
Variations	**2b**: The user has a choice of file type but can only select midi for now.
Non-functional	none
Issues	Possible loss of some data such as chord tracks.
Exceptions	If User-A has non-midi tracks, a dialog needs to appear warning them that the audio tracks will not be exported.

Use Case	**Import a Composition from a Midi File**
Description	A user wants to load a midi file as a composition in their library and start working on it immediately
Actors	User-A – The person importing their midi composition
Assumptions	• User-A has a midi file ready to be imported.
Steps	1. User-A actions to import. 2. A dialog appears asking them to select the midi file to import. 3. The file is imported and opened as a composition with file name (without the extension) as the composition title.
Variations	none
Non-functional	none
Issues	none
Exceptions	none

3. Playback Controls



Use Case	**Control Audio Playback of Compositions**
Description	A user wants to control real-time playback of a composition.
Actors	User-A – The one playing back the composition
Assumptions	User-A has a composition open and the playback marker is not at the end.
Steps	1. User-A actions to play and hears the composition play back through his speakers. 2. User-A actions to stop. The audio is instantly responsive and stops. The playback marker returns to the beginning of the composition. 3. User-A actions to skip to end. The playback marker moves to the end of the composition (after the last note). If 2c variation, then this step has no effect. 4. User-A actions to rewind to start, and the playback marker returns to the start of the composition. 5. User-A actions to fast-forward and the playback marker is now at 8 seconds.
Variations	**2b**: User-A actions pause. The audio is instantly responsive and stops. The playback marker remains where it stopped. **2c**: The playback marker reaches the end of the composition, and pauses automatically.
Non-functional	**Performance**: All actions need to be highly responsive.
Issues	• With so many controls the user might get confused and not know what button does what. Client has proposed text on each playback control as well as a meta-mode. • Should clicking stop return the playback marker to the position it was in when the user clicked play rather than to the beginning of the entire composition?
Exceptions	User-A is deaf like Beethoven. In that case, User-A has no need for the playback controls at all.

4. User Accounts

Use Case	**Registering a New Online Account**
Description	A user wants to have the ability to share their compositions
Actors	User - The one who will get a new account
Assumptions	The user has just started the application and they are connected to the internet.
Steps	1. When presented the option to login, User opts instead to register a new account. 2. User is then prompted for a username and other information for their new account. 3. User decides whether they want to be searchable by other online users. 4. User is then shown all their selection choices, and prompted to confirm all is correct. 5. The system then creates the account and User is able to use online mode
Variations	**2c:** If the username is already taken, the user is prompted to change it to a new, unique one.
Non-functional	None
Issues	We need an obvious help describing what each selection means e.g. being searchable by other users.
Exceptions	

Use Case	**Deleting an Account**
Description	User wishes to delete his/her account.
Actors	User - The one who owns the account to be deleted.
Assumptions	User is in online mode (logged into their account they wish to delete)
Steps	1. User selects to delete their account. 2. User is prompted for and enters a password, for authentication. 3. User is warned that these actions are permanent, and is shown choices again for confirmation. 4. User confirms the actions, and the account is deleted.
Variations	None
Non-Functional	None
Issues	None
Exceptions	If the password provided is incorrect then the user is prompted to re-enter the password.

Use Case	**Entering and leaving online mode**
Description	User wishes to enter online mode, then exit online mode.
Actors	User - The one using the program
Assumptions	The user has just started the program and is not in online mode. The user is connected to the internet.
Steps	1. The user actions to enter online mode 2. User enters their credentials. 3. User is now in online mode and proceeds to join chat rooms, have conversations, and share compositions, as well as editing compositions. 4. When finished using the online-mode functionality, the user exits online mode and continues to work on their composition.
Variations	If the user enters invalid credentials then the user is asked to re-enter the invalid credentials.
Non-functional	None
Issues	None
Exceptions	None

5. Online Communication

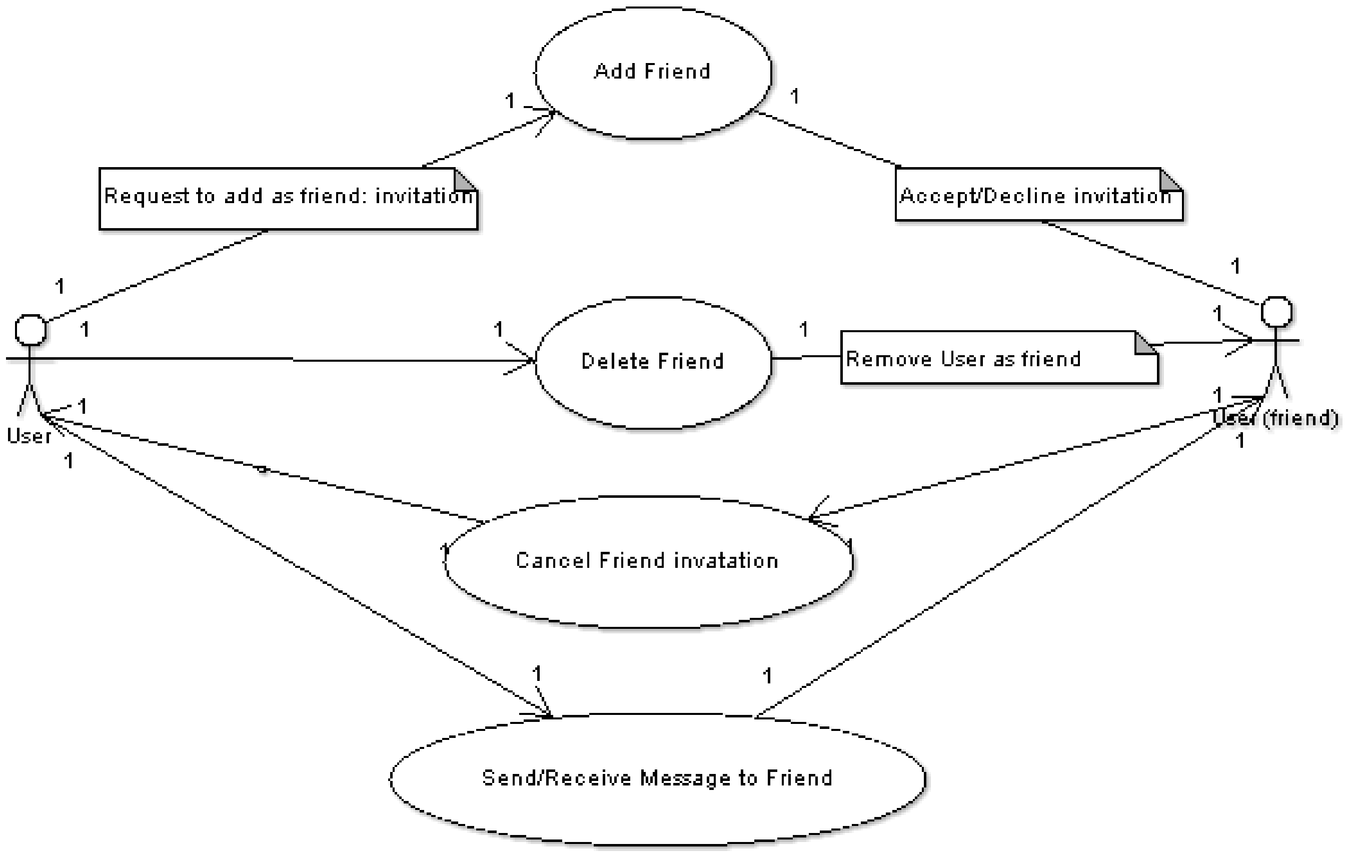
Use Case	**Select & Enter Chat Room**
Description	A user wants to enter in a chat room to communicate with other composers.
Actors	User-A – The one entering the chat room.
Assumptions	User-A has logged on their account (online mode).
Steps	1. User-A opens chat-room-listings. 2. User-A can view all of the currently available chat-rooms. 3. User-A takes a long time to choose a chat-room, so the user refreshes the chat-room listings to update the information. 4. User-A selects a listed chat room, the information of that selected chat-room is refreshed. 5. User-A joins/enters the selected chat-room. 6. A chat-room opens up and they have now entered the chat-room.
Variations	**2b**: The user can search the listed chat-rooms.
Non-functional	**Response**: The retrieving and displaying of chat-room listings should have a fast response time.
Issues	• Should the chat-room-listings be opened automatically and always stay open once logged-in? • Should the window be closable or hide-able? • Should the listings be refreshed automatically or manually? • Synchronization, threading, simulating/testing will be difficult.
Exceptions	User-A is disconnected (for some external reason). User-A is notified that they have been automatically logged-off and switched to offline mode. The chat-room listings disappear, any chat-rooms that the user has connected to disappear.

Use Case	**Start a New Private Conversation**
Description	User wants to start a new conversation to communicate in private with other users
Actors	User-A - The one who wishes to start the private conversation. User-B - The one whom User-A wishes to converse with.
Assumptions	User-A has logged in and connected to online mode. User-B is also connected to online mode.
Steps	1. User-A selects another user with which they wish to start a new private conversation (User-B). 2. User-A sends an invitation to User-B to start a new private conversation. 3. User-B accepts, and a new room is set up. User-A and User-B can now chat in private.
Variations	If User-B does not wish to chat with User-A, they may decline the invitation, in which case User-A is notified and no chat is created.
Non-functional	None
Issues	• Where will the new room be based? • If User-A leaves the private room, inform User-B.
Exceptions	None

Use Case	**Communicating in a Conversation**
Description	User wants to communicate with other users in a conversation
Actors	User - The one who wishes to send messages
Assumptions	User has logged in and connected to online mode. User has joined a conversation, either public or private
Steps	1. User sends a message to all other users in the conversation by simply typing his/her message and pressing [Enter]. 2. User sees a copy of the transmitted message, as do all other users in the conversation. 3. User selects another user, and selects to send a private message. 4. User's message is then only transmitted to that singular user, and is not seen by anyone else in the conversation.
Variations	**3b-4b:** When sending a private message, User may, instead of selecting the recipient in the room list, enter a special message prefix which will indicate that the message is private and to be sent to only the named recipient.
Non-functional	Messages must be received by all users as quickly as possible after being sent (~100's of milliseconds)
Issues	None
Exceptions	None

Use Case	**Inviting a User to a Private Conversation**
Description	User-A wishes to invite User-B to an existing private conversation of which User-A is already a member.
Actors	User-A - The user already in the conversation who wishes to allow User-B to join. User-B - The user who will be invited to the private conversation.
Assumptions	Both User-A and User-B are logged in and connected to online mode. User-A is connected to an existing private conversation.
Steps	1. User-A selects the user they wish to invite to the existing private conversation. 2. User-A then extends an invitation to User-B, offering to let them join the conversation. 3. User-B accepts the invitation, and is added to the private conversation. 4. All other users in the conversation are notified that a new user has joined the conversation, and who invited them.
Variations	**1b:** The methods User-A can use to select User-B vary, including selecting from a room list, typing in their username, or doing a search. **3b:** If User-B declines the invitation, User-A is notified, and User-B is not added to the conversation.
Non-functional	None
Issues	When User-A is inviting a User-B to a private room, when User-A is involved in multiple private conversations, need a way to indicate to which room User-B is being invited into.
Exceptions	None

6. Friends


Add Friend
Request to add as friend: invitation
Accept/Decline invitation
Delete Friend
Remove User as friend
Cancel Friend invatation
Send/Receive Message to Friend
User
User (friend)
1

Use Case	**Add Friend**
Description	A user wants to add a friend to their friends list
Actors	User-A – The user who wants to add a friend. Username: “John” User-B – The user who user-A wants to add. Username: “Bob”
Assumptions	User-A and User-B have logged on their accounts (online mode) User-A and User-B are not on each others friends list.
Steps	1. User-A opens the friend-finder. 2. User-A **searches** for a friend by attributes such as age, genre, country etc. Listings can be ordered by any attribute. 3. User-A selects User-B, whom they wish to add as a friend. 4. User-A clicks an add-friend button. 5. User-B’s username (“Bob”) is now displayed on User-A’s friends list, but its status is set as “waiting for invitation acceptance”. Users listed with this status aren’t yet friends. 6. User-B receives an invitation to become a friend with User-A. 7. User-B chooses the accept option. Before this decision User-B can view User-A’s profile. 8. User-A and User-B now listed on each others friends list.
Variations	**2b**: User-A **browses** for a friend in the friend-finder by genre or instrument. **3b**: User-A not only selects but opens up the users profile, showing details omitted from the search/browse results listings. **2 - 4b**: User-A manually adds a friend via typing a username. **7b**: User-B declines the invitation. User-A receives notification that the invitation has been declined. **7c**: User-B selects cancel to postpone making the decision to accept the invitation. User-B can bring back the invitation either via there friends list (“requesting to be added friend”) or it pops up when they next log-in. In the meantime User-A waits.
Non-functional	None
Issues	• Some users don’t want to be listed in the friend finder (for privacy reasons). The option to be unsearchable should be an option which can be toggled when a user is online. • Synchronization, threading, simulating/testing will be difficult.
Exceptions	• A user tries to add a friend who is already a friend. The user is notified and nothing else happens. • A user tries to manually add a username which does not exist i.e. a user who closed their account prior to the adding. The user who is attempting to add the user as a friend is notified. Nothing else happens. • User-A goes offline but invitation still hasn’t been accepted. The invitation is still active and User-B can accept it whenever the wish (until User-A cancels it). When User-A next comes online they will still see whether or not that User-B has accepted the invitation or not. Can be: Status can be waiting for acceptance, accepted or declined.

Use Case	Delete Friend
Description	A user wants to delete a friend on their friends list
Actors	User-A – The user who wants to delete a friend (User-B) User-B – The user who User-A is deleting
Assumptions	User-A and User-B has logged on their account (online mode) User-A and User-B are accepted friends.
Steps	1. User-A selects User-B on their friends list. 2. User-A actions to delete the selected friend. 3. User-A is presented with a confirmation message & confirms. 4. User-B is removed from User-A's friends list. 5. User-B is notified that User-A has deleted them as a friend and User-A is automatically removed from User-B's friends list.
Variations	**3b**: User-A cancels the operation. Then nothing else happens.
Non-functional	None
Issues	• Synchronization, threading, simulating/testing will be difficult. • The friend-ship can be restored via the adding friend process.
Exceptions	User-A deletes User-B (who is offline) from their friends list. In this case, step 5 occurs when User-B next comes online.

Use Case	Cancel Friend Invitation
Description	A user who is waiting for another user to accept a friend invitation and decides to cancel the invitation.
Actors	User-A – The user who wants to cancel friend invitation. User-B – The user that received a friend invitation from User A.
Assumptions	User-A and User B are logged on their account (online mode). User-A has sent a friend invitation to User-B which has not yet been accepted.
Steps	1. User-A selects to cancel the invitation action for User-B to be added as a friend. 2. A confirmation message pops up and User-A confirms. 3. User-B is removed from User-A's friends list. 4. User-A's invitation disappears from user-B.
Variations	**3b**: User-A cancels the operation. Nothing else happens
Non-functional	None
Issues	Synchronization, threading, simulating/testing will be difficult.
Exceptions	• If User-B is viewing the invitation when User-A cancels it, User-B should be notified before the invitation window closes. • User-B accepts after User-A cancels the invitation (possible due to delay). If the cancellation request reaches the "communication-management-system" first then User-B's accept will be discarded and User-B will be notified of the cancellation. If User-B's accept arrives first then the friendship is established and User-A will be notified that User-B has already accepted the friend invitation.

Use Case	**Communicate with a Friend**
Description	A user wants to send and receive messages, with a friend.
Actors	User-A – A user wanting to send a message to user-B User-B – A user receiving messages from User-A
Assumptions	User-A and User-B are logged onto their accounts (online mode). User-A and User-B are accepted friends.
Steps	1. User-A selects User-B from their friends list. 2. User-A starts a conversation with the selected user. User-B knows nothing of the conversation at this point. 3. User-A sends the message "Hello" in the conversation. 4. User-B sees the conversation with the message ("Hello") from User-A. 5. User-B replies and sends a messages to User-A. 6. The private conversation continues until one of the users chooses to end the conversation.
Variations	**General**: If User-A is already is in a public conversation with User-B they can send a message via that conversation instead.
Non-functional	**Response**: Messages should be instant.
Issues	• How should User-B be notified without being interrupted? Especially when user-B is in the process of composing. • Synchronization, threading, simulating/testing will be difficult. • Users should not be notified while in record mode.
Exceptions	• User-A's message can not be sent to User-B for some external reason. User-A must be notified in a reasonable amount of time that the message was not sent, and if possible provide useful information on what is wrong and how they can fix it. • User-A is disconnected once they have sent the message. User-B receives the message and the conversation window still appears, User-B is notified that the conversation has finished and that they cannot send a message back to User-A. • User-B disconnects while User-A sends the message (i.e. just before so that the assumptions hold). User-A is then notified that the message send failed because User-B is now offline. • User-B deletes User-A from their friends list after the initial message was sent but just before the message is received by User-B. This can have 1 of 2 outcomes: Either the process continues as normal or User-A is notified that their message could not be sent because they are not long friends with User-B (They could then have an option to send an invitation to User-B to enter a private conversation).

7. Sharing Compositions

Use Case	**Send composition to user**
Description	A user wants to send a full composition to another user
Actors	User-A – The user who wants to send their composition to User-A. User-B – The user who is to receive the composition.
Assumptions	User-A and User-B have logged on their accounts (online mode).
Steps	1. User-A selects the composition they wish to share with User-B 2. User-A drags the composition over to User-B (who is displayed in friends list or in a public/private conversation). 3. Sharing confirmation message is displayed & User-A confirms. 4. User-A waits for User-B to accept the invitation to download the composition and can continue working while waiting. 5. User-B receives the invitation and chooses to accept. 6. Both users can now see the progress. 7. A period of time elapses. During this time the users work on other tasks. 8. Once the download/upload has finished the users are notified. 9. User-B now has the composition stored in their library. The location of the newly downloaded composition is clearly made known to User-B, who is given the options: “Play composition”, “Edit Composition”, “Organize Composition” and “Cancel”.
Variations	**3b**: User-A decides to cancel the operation. Both users are notified that the download is cancelled and the process is stopped. **4b**: User-A cancels the invitation. This ends the operation. **5b**: User-B declines the invitation. User-A is notified and the operation is aborted. **7b**. One of the users cancels the download during the downloading / uploading time. The other user is notified and the whole operation is aborted. All downloading and uploading progress is lost. **Send-Single track:** A similar use-case, except instead of sending a full composition, a single track inside a composition is shared.
Non-functional	None
Issues	• Synchronization, threading, simulating/testing will be difficult. • Maybe eventually look at resume-able/recoverable composition sharing to avoid wasted downloading & uploading times.
Exceptions	**4**: User-A disconnects / goes offline while waiting for User-B to accept the invitation. The invitation is then cancelled. **4b**: User-B accepts the invitation to share after User-A cancels the invitation (the cancel isn’t yet known by User-B). In this case User-B will be notified that the composition-share was cancelled and user-A will not be aware of User-B’s acceptance. **7:** One of the users goes offline during the sharing. The operation is cancelled and all sharing progress is lost.

8. Track Generation

Use Case	Creating a New Track
Description	The user wants to add a track to a composition and view it.
Actors	User – the composer who is creating the track.
Assumptions	The software is open with a composition loaded.
Steps	1. The user selects the 'add track' option, specifying the type of track to add (chord, beat or note). 2. The user selects the instrument to play the track. 3. The user enters a name to uniquely identify the track. 4. Empty track is displayed in expanded/edit mode.
Variations	None
Non-functional	None
Issues	• Should User be forced to specify a unique name for the track? • Should a single dialog box be displayed when the user clicks add track which allows the user to specify track options.
Exceptions	None

Use Case	Deleting a Track
Description	The user wants to permanently remove an unneeded track.
Actors	User – the composer who wishes to remove a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another unneeded music track.
Steps	1. The user selects the track to delete. 2. The user selects the delete track option. 3. The track is removed from view.
Variations	None.
Non-functional	None.
Issues	If the user can undo track deletions they must be prompted to confirm the permanent track deletion.
Exceptions	None.

Use Case	Duplicating a Track
Description	User wants to use an existing track as the base for a new track.
Actors	User – the composer who wishes to duplicate a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track.
Steps	1. The user selects the track to duplicate. 2. The user selects the duplicate track option. 3. The entire track is duplicated.
Variations	None.
Non-functional	None.
Issues	Should the user be prompted to specify a name for the duplicate?
Exceptions	None.

Use Case	**Adding Music to Tracks**
Description	The user wants to add music to an existing track.
Actors	User – the composer who is adding music to a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track with 6 bars of music.
Steps	1. The user moves the current position to bar 4. 2. The user adds 3 bars of music in normal mode. The track now contains 9 bars of music. 3. The user then switches to overwrite mode and enters another 3 bars of music. 4. The final three bars are overwritten and the track still contains 9 bars of music.
Variations	None.
Non-functional	None.
Issues	• User the Word processor model to allow quick entry of notes into a track. Aid users in learning the software. • User must be able to undo the actions of adding notes.
Exceptions	None.

Use Case	**Modifying Track Selections**
Description	The user wants to modify an existing track.
Actors	User – the composer who is modifying a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track with 6 bars of music.
Steps	1. User selects the first two bars and sets volume to loud. 2. User selects the final two bars and sets volume to quiet. 3. User selects the whole track and adds a reverb effect. 4. User selects the middle 4 bars and plays them back in looped playback mode to check that the addition of dynamics and effects has had the desired affect.
Variations	None.
Non-functional	None.
Issues	• User must be able to undo all these modifications of selections. • Display must update immediately to appropriately reflect these changes.
Exceptions	None.

Use Case	**Cut, Copy, Paste**
Description	The user wants to copy and paste sections of a track
Actors	User – the composer who is editing the track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track with 6 bars of music.
Steps	1. The user selects the first 3 bars of music in the track. 2. The user chooses to copy the selection. 3. The user then moves the cursor to the end of the track and pastes the copied music. 4. The composition now contains 9 bars.
Variations	**2b**: The user cuts the selected music then pastes it at the end. The resulting track will have what were originally the first three bars now at the end. **3b**: The user chooses to paste the copied music at bar 4 in overwrite mode. The three copied bars overwrite the existing music and the track now contains 6 bars, of which the last 3 are a duplicate of the first 3. **3c**: The user chooses to paste the copied music in normal mode at bar 4. The track now contains 9 bars of which the first 3 and last three bars are identical.
Non-functional	None
Issues	• Make sure it is clear what is selected with visual feedback. • Follow the Word processor model. In overwrite mode pasting overwrites existing music, but in normal mode pasting inserts the copied music.
Exceptions	None

Use Case	**Deleting and Inserting Rests**
Description	The user wants to delete sections of a track.
Actors	User – the composer who is editing the track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track with 9 bars of music with no rests.
Steps	1. User selects the first 3 bars of music in the track. 2. User chooses the silence option. The track now contains 9 bars with the first 3 bars rests. The 3 bars remain selected. 3. User chooses the delete option to remove the 3 bars. 4. User sets the current track position to bar 4. 5. User inserts 3 bars of rests. 6. The resulting track contains 3 bars music at the beginning and end with 3 bars of rests in between.
Variations	None.
Non-functional	None.
Issues	• Make sure it is clear what is selected with visual feedback. • Following word processor model use space bar for adding rests and insert key to toggle overwrite, add and normal mode. • User must be able to undo the action of deleting notes.
Exceptions	None.

9. Track Display

Use Case	**Viewing and Printing a Track**
Description	The user wants to view and existing track and print it
Actors	User – the composer who is displaying and printing a track.
Assumptions	The software is open with a composition loaded which contains at least 1 note track with note information.
Steps	1. User selects the note track they wish to view and print. 2. The track is displayed on screen. 3. User selects the print option to print the track. 4. The track prints out on a sheet of paper.
Variations	**3b:** User chooses the print preview option to check and set various options for the look of the document before printing. **3c:** User chooses to print the entire composition resulting in all tracks being printed at once.
Non-functional	Print outs must look neat and tidy, especially when there are more than one track being printed out at once.
Issues	• Need to make sure tracks are printed large enough to see. • When user selects to print all tracks, print them out one at a time – not on the same sheet.
Exceptions	None

Use Case	**Track Navigation**
Description	The user wants to quickly view each bar in the track from the start as well as listen to specific segments.
Actors	User – the composer who is adding music to a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track with 60 bars of music. The current position is in the middle of the song.
Steps	1. User can skips to the start of the track in a single action. 2. User then selects bars 1 to 4 and listens to them in normal playback mode. Playback ends when the selection is finished. 3. User then skips forward two pages of music and listens two the next 4 bars of music. 4. User then skips to the end of the composition in a single action. 5. User zooms in to get a better view of the details in the final page of the track. 6. Finally user selects the final page of music and listens to it.
Variations	**2b:** The user could listen looped playback to repeatedly play the selection until the user clicks stop. **5b:** User zooms out to view more of the final section of the song on the screen at one time.
Non-functional	None
Issues	Use the word processor model for navigation shortcuts.
Exceptions	None

10.Master Track

Use Case	**Adding Chords to the Master Track**
Description	User wants to add chords to a track using a particular key
Actors	User – the composer who is adding chords to the master track.
Assumptions	The software is open with a composition loaded and the user has chosen to add a new chord track.
Steps	1. User selects a key from a list of available keys. The chords for the selected key are shown and can be added to the current track with a single action. 2. User adds some chords to the current track then selects a new key. 3. The available chords change to reflect the new key. This has no effect on the track and the recently used chords.
Variations	None
Non-functional	None
Issues	Could end up with a lot of chords being displayed if we have both recently used chords and chords from a key displayed.
Exceptions	None

Use Case	**Copy and Paste**
Description	The user wants to copy and paste sections of a master track
Actors	User – the composer who is editing the master track.
Assumptions	The software is open with a composition loaded. The master track already contains 3 chords, each 1 bar in length.
Steps	1. User selects all 3 chords. 2. User chooses to copy the selected chords. 3. User then selects the next empty bar and pastes the copied chords. 4. The composition now contains 6 bars
Variations	None
Non-functional	None
Issues	• Need to make sure it is clear what is selected with good visual feedback. • Pasting in a bar that already has chords in it overwrites the existing chords. Selecting insert causes the bars to be inserted and everything else to shuffle along.
Exceptions	None

Use Case	**Chord Composition**
Description	Modifying a master track using the recently used chords feature
Actors	User – the composer who is adding strums to a chord track.
Assumptions	The software is open with a new composition containing a blank master track.
Steps	1. Initially there are no recently used chords visible. 2. User selects a chord from a list of chords and adds it to the track. This chord is added to a list of recently used chords at the same time. 3. User selects a different chord and adds it to the track. The 2nd chord is added to the front of the most recently used list. 4. User drags one of the chords from the recently used chords list onto the track.
Variations	None
Non-functional	None
Issues	Once the recently used chord list is full, adding a new chord to the track will result in the replacement of the chord at the end of the most recently used list.
Exceptions	None

Use Case	**Setting Tempo, Key, Style and Volume**
Description	Modify the attributes of the composition master track.
Actors	User – the composer who is modifying the master track.
Assumptions	The software is open with a composition containing an imported midi file.
Steps	1. User sets the track volume to 80%. 2. User selects the style for the composition from a list. 3. User selects the initial key for the composition from a list. 4. User enters 80 as the new initial tempo for the composition.
Variations	**3b:** User selects and existing midi track and the 'key analyser tool' which determines the song key, and requests confirmation to save as the song key. **4b:** User changes the tempo by selecting the 'tempo setting tool' and by tapping a designated key with the desired timing.
Non-functional	None.
Issues	None.
Exceptions	If the song contains key changes the user is informed of the initial key as well as the locations of key changes.

11.Chord Track

Use Case	**Creating a chord track (Strum patterns)**
Description	User wants to have a specific guitar strum pattern in a chord track.
Actors	User – the composer selecting the guitar strum pattern.
Assumptions	The software is open with a composition loaded and the user has added chords to the master track added a new chord track.
Steps	1. User selects guitar as the new chord track's instrument. 2. User selects the guitar strum pattern to be used for each bar in the chord track.
Variations	**2b:** User selects a bar that already has a specified strum pattern and changes to a different strum pattern.
Non-functional	None.
Issues	• There should be some visual indication of the rhythm of the strum when looking at the track. • The list of strums is dependant on both the chord track instrument as well as the time signature.
Exceptions	None.

Use Case	**Chord Track Composition**
Description	Modifying a chord track using the recently used strums feature
Actors	User – the composer adding strums to a chord track.
Assumptions	The software is open with a composition containing a blank chord track and a master track with chords filled in.
Steps	1. Initially there are no recently used chords visible. 2. User selects a strum pattern from a list and adds it to the track. The strum pattern is added to the most recently used strum list. 3. User selects a different strum and adds it to the track. The second strum is added to the top of the most recently used list. 4. User drags one of the strums from the recently used strums list onto the track.
Variations	None
Non-functional	None
Issues	Once the recently used strums box is full, adding a new strum will result in the removal of the strum at the bottom of the list.
Exceptions	None

12.Note Track

Use Case	**Adding Music to Tracks**
Description	The user wants to add piano music to a new note track.
Actors	User – the composer who is adding music to a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and a blank note track.
Steps	1. User sets the note track instrument to piano. 2. User clicks in 8 bars of single notes in the treble clef. 3. User enters add mode and adds in a second part in the right hand. All added notes have the same lengths as the notes added in step 2. 4. User then adds 8 bars of a combination of chords and arpeggio's in the left hand. 5. User then adds accents and staccato's to particular notes for emphasis. 6. User adds a crescendo in bar 3. 7. User concludes by adding a piano dynamic for the first for bars and forte dynamic after 4 bars.
Variations	None.
Non-functional	None.
Issues	• User the Word processor model to allow quick entry of notes into a track. Aid users in learning the software. • User must be able to undo the actions of adding notes.
Exceptions	None.

Use Case	**Modifying Notes in a Track**
Description	User wants to modify the notes in an existing track.
Actors	User – the composer who is adding music to a track.
Assumptions	The software is open with a composition loaded. The composition contains a master track and another music track with 6 bars of music.
Steps	1. User selects the entire track and shifts the notes up an octave. 2. User selects the first two bars in the bass clef and shifts them down an octave. 3. User selects the first bar in the treble clef and adds a slur. 4. User adds a decrescendo to the final two bars. 5. User adds a forte dynamics at the opening bar.
Variations	**1b:** User can select all the notes on one stave rather than the entire track and shift them down or up and octave.
Non-functional	None.
Issues	• User the Word processor model to allow quick entry of notes into a track. Aid users in learning the software. • User must be able to undo the actions of modifying notes.
Exceptions	None.

13.Beat Track

Add Frances Use Cases here…

14.Composition Tools

Use Case	**Chord Analyser**
Description	The user wants to automatically generate the chords in the master track based on a piano composition. The user wants to check and modify the chord analysers suggested chords.
Actors	User – the composer who wants to use the chord analyser.
Assumptions	The software is open with a composition containing a piano track imported from a midi file or recorded from a midi keyboard.
Steps	1. User selects the piano track to be analysed. 2. User duplicates the piano track. 3. User runs the quantize tool on the duplicate of the piano track. 4. User selects the duplicate track. 5. User runs the key analyser and sets the composition key. 6. User specifies chords to be included and excluded from the analysis. 7. User runs the chord analysing tool. 8. User created a chord track for a brass section with the default rhythm pattern. 9. User expands the view of the master track. 10. User plays back the chord analysis with the piano track while viewing the master track contents. 11. User enters highlight mode. 12. User highlights chords that it does not like during playback. 13. User turns off highlight mode. 14. User manually changes chords it didn't like.
Variations	**1b:** In a composition containing multiple note tracks the user may specify one or more tracks to be used in the analysis. **2b:** User can also specify the style to be used in the analysis. **4b:** User can specify to receive two separate predicted chord progressions to listen to and compare and eventually merge into one. This is particularly useful if little note information is available in the note tracks.
Non-functional	None.
Issues	• What happens if the song already contains chord information in the master track? The user could be the original line of chords displayed above the chord analyser's chords and forced to select the best chord where conflicts arise. • Should quantization be performed automatically on a copy of the tracks selected to be used in the analysis?
Exceptions	Where limited chord information was available in the track used for analysis, predicted chords are identified as such. For predicted chords the user can request another prediction if it did not like the initial prediction.

Use Case	**Composing by Song Phrase**
Description	The user wants to compose piano composition a phrase at a time and decide on the structure after composing each of the phrases.
Actors	User – the composer who wants to compose song phrases.
Assumptions	The software is open with a new composition an empty master track.
Steps	1. User adds a new piano note track. 2. User selects create new phrase and titles it 'Chorus'. 3. User adds the piano notes to the chorus phrase. 4. User selects create new phrase and titles it 'Verse''. 5. User adds the piano notes to the verse phrase. 6. User selects create new phrase and titles it 'Bridge''. 7. User adds the piano notes to the bridge phrase. 8. User selects create new phrase and titles it 'Intro''. 9. User adds the piano notes to the intro phrase. 10. User selects create new phrase and titles it 'Ending''. 11. User adds the piano notes to the ending phrase. 12. User specifies the song structure as intro, verse, chorus, verse, chorus, bridge, chorus, ending. 13. User listens to the song and decides to remove the second last chorus because the song is too long. 14. User selects the option to view and edit the chorus track and makes improvements. 15. User then renames the verse phrase 'verse1' and duplicates it. The duplicate is called 'verse2'. 16. User makes slight modifications to the duplicate of verse2 to add variety.
Variations	**2b:** User could add several tracks here. Every phrase in the composition must have a same set of tracks. So adding a track to one phrase adds the track to all the other phrases.
Non-functional	None.
Issues	• Is there anything wrong with forcing users to give unique titles to their phrases? • It would be very nice but probably complicated if specified tracks in duplicated phrases could be linked so that modifications to the track in one phrase were reflected in the linked track in the original phrase i.e. the drum tracks for a verse1 and verse2 tracks could be the linked but the piano tracks could be modified and recorded separately for in each phrase. This could start to get a little confusing.
Exceptions	None.

15.Help

Worry about help system later in the project, due to the fact we only have 12 weeks to complete the project. In reality help and documentation would be given slightly higher priority.