PRCO304 - Project Initiation Document
Online Real-Time Music Collaboration Tool

1. Introduction

Currently, most music production software is a native application, that being, that it does not run on or even require the Internet. The benefits of this is that it allows to be extremely powerful, allowing the software to make full advantage of the computer's capabilities. They often have in-built synthesisers, drum machines and effects units; along with the ability of being able to plug in external machines it allows the client to produce and create any imaginable genre of music. This gave power to the individuals, now anyone with an idea and willingness can learn and create music, regardless of age, money or training. However, even though modern applications strive to make the learning curve as minimal as possible, the use of them to a beginner can be quite daunting. This and the fact that it is near impossible to be able to collaborate on a single music project in real-time unless in the same room give rise to a few, but very important, disadvantages.

The rise of the Internet saw a slow and steady increase in web-based applications. Only recently with the advancements and wide acceptance of JavaScript and its in-built capabilities of creating manipulating sound has it really began to move forward. There are a few powerful, in-depth web-based music production tools and a whole host of simple, standalone, web-based synthesisers, drum machines and effect units. However, very few, if not none, make full advantage of the raw capabilities of the internet, communication. I believe there is room for a web-based music production tool that allows for real-time collaboration. The only requirements a user needs is access to the Internet, minimal knowledge of music, how a sequencer works (although should be intuitive enough to pick up without prior knowledge) and someone to create music with.

2. Business case

2.1 Background and Motivation

Music production software has taken many forms since the inception of the computer. They allow clients to write, produce and create all different genres of music, only needing a computer and the software. Modern adaptations strive to make it as easy as possible, with little-to-none prior knowledge of music needed to get going and a wealth of in-built synthesisers and drum machines, often replicating real world tools to a high standard. They can be used in conjunction with external musical hardware such as synthesisers or drum machines by manipulating MIDI signals and are powerful enough to allow the expert user to create best selling albums. However, professional software is very costly and requires a relatively powerful computer to operate effectively on. They also are very solo artist orientated. That meaning, unless in the same room as the producer, it would be near impossible collaborate on the project.

Only recently with the wider acceptance of JavaScript with its in-built Web Audio API capabilities that allow for creation and manipulate sound, has web- based music applications have risen. The advantages of having web-based music creation software over bespoke native applications is the same as the advantages of the Internet in general. Anyone with access to the internet can create music as well as allowing collaboration on a worldwide scale. The amount of web-based production software, synthesisers, drum machines has steadily been increasing. Most however, are still standalone, that focus on a particular genre or hardware and aren't multi-user focussed. I see that there is a real opportunity to create some music production software that allows for real-time collaboration. This meaning that not only is the access to the software is only restricted by the user having access to the internet, it also allows for two people, from anywhere in the world, to collaborate on the same music project at the same time. It should be intuitive enough for basic users to be able to use it efficiently and have the capabilities for advance users to be able to create and manipulate to a high degree.

2.2 Business Objectives

1. To allow for collaboration on music projects with two or more people.
2. Remove the need to send music projects back and forth or to have to meet up in 'real-life' to collaborate on a music project together.
3. To have a basic web-based digital audio workstation for creation of music in with the use of external hardware if wanted.
4. To be able to intuitively use the software to create a piece of music without much prior knowledge of how to do so.
5. To be able to track, manage and edit created pieces of music.
6. Remove the need for an expensive, computationally heavy piece of software to achieve this.

3. Project objectives

1. Analyse and current music creation process with music with current musical software to understand the basic needs of musicians working with computers.
2. User portal to be able to keep information about the user and for them to be able to manage, edit and delete their music projects.
3. A web-based digital audio workstation that allows for creation of scores of music.
4. A web-based digital audio workstation has pre-loaded samples that they can use.
5. Two or more users can simultaneously work on the same project in real-time and see the other's inputs reflected on the project as they happen.
6. They can manipulate the sound parameters of the samples.
7. Users can then download their musical composition as an audio file.
8. An intuitive interface that allows for quick and easy music creation.

4. Initial scope

I have decided to use the MoSCoW technique in order to measurably define my project requirements. This ranking technique is defined by: M - must have, S - should have, C - could have, W - would have. In order for my project to be defined as successful, it should have met all ’Must have’ defined requirements. The numbering in the table below refers to the same number of that requirement defined in the ’3. Project objectives’ as to expand information on that specific requirement, and any further sub-points will be defined alphabetically.

Initial Scope

No.	Requirement	Priority
1.	The process of music creation will be identified via interviews and observations and documented appropriately.	M
a.	They will then be further analysed and user stories, UI designs and user requirements will be drawn from it.	M
2.	The user will be log onto their own web-portal in the system that tracks their created and worked upon projects.	M
a.	They will be able to delete projects that they’ve worked upon.	S
b.	They will be able to further edit projects that they’ve worked upon.	C
3.	Two or more users will be able to access a page with a music sequencer that has basic step sequencer to allow for ‘click-to-create’ musical patterns.	M
a.	More step sequencers should be able to be added and deleted from the project to allow for more layering of sounds as the user’s see fit.	C
b.	The music that is created with external instruments through MIDI implementation (IF implemented) will be represented as a score or piano roll.	S
4.	Database stored samples that the users can use to create the music, ranging from drum, piano and synthesiser samples.	M
a.	The user should be able to upload their own samples to be able to use in the project.	C
b.	The users should be able to manipulate the samples using an external device connected to the computer and controlled by MIDI signals.	C
5.	The users can see each others inputs to the project as and when they happen to create the real-time effect.	M
a.	The project shouldn’t be limited to amount of people working on it at the same time.	S

b.	The users can communicate to each other through an in-built messaging service.	C
c.	Each user can save the a version of the project.	M
6.	The sound parameters of the sample, such as attack, decay, sustain and release can be editted by the users.	S
a.	These parameters of the sample are recovered if the user later goes to edit the song	W
b.	Further effects such as delay, compression, reverb can manipulate the sound further	W
7.	The project that the user’s have created can be downloaded as an MP3 or WAV file to be played as they wish later.	S
8.	UI is intuitive enough for user’s of minimal skill level to be able to successfully produce music as well as being in-depth enough to serve the need of more advanced users.	M
a.	User feedback will be taken to ensure this is achieved regularly throughout the project and documented accordingly.	M

5. Resources and dependencies

The project isn’t currently dependable on any external resources or products. I have provided the server in which it is run which is funded by myself which has all capabilities to host application and database.

I may use the front-end framework ’Express’[1] that is heavily used throughout many NodeJS projects, however this is isn’t decided for certain yet and even if so, the project is very stable and with a large following so it wouldn’t cause any problems. I am also planning to use the third-party library socket.io[2] to help with the real-time client to server communication. This is because it helps with ease of development and it isn’t realistic to create everything myself when suitable alternatives are available. Both frameworks are highly recognised and have a big community surrounding them so they shouldn’t cause problems in their failure to be maintained.

Potentially, for some desirable features, I might use other pre-existing JavaScript library to make development faster. However, I intend to avoid the use of them at all cost throughout my project. If such libraries are used, a document of how they are used and the justification for using will be in the final report as well as weekly summary.

[1] http://expressjs.com/

[2] http://socket.io

6. Method of approach

After some research I have decided my software development process will follow that of an AGILE approach using SCRUM methods. That is, to release working minimal viable products early and often. This is because I have experience with this process, I know how to plan the work and I would like to get minimal viable products out as soon as possible. For testing and feedback purposes.

The first stage of iterations will be to get the core product working and usable. I want a step-sequencer that a user can create simply pattern and melody using basic samples. As well as a having the real-time aspect working to a simple standard. A simple login system to store user's details.

The second stage of iterations will be to enhance the web-portal for the users, including viewing of tracks worked upon and the ability to manage said tracks. The musical compositional tool

Once these are in place, that is the core functionality of my web-application done and I would like to focus on incrementally improving the product and adding desirable features as noted in section 4, anything that isn't a an M in the MoSCoW ranking technique.

Possible technologies are NodeJS and within the JavaScript Web Audio API, Express framework and Socket.io library. Also HTML and CSS for all front end styling and layout. I haven't decided whether I will be using a relational database or a NoSQL database yet for storage of samples and potential storage for user's session/work. I need to revise and my data that will be stored and see which suits best. I'm leaning towards a NoSQL approach due to the fact the data will be of different types and if I am storing user sessions the data will be unstructured or at least difficult to restrain into a table. If I do decide to go with a relational database I will use MySQL as I've had previous experience with it, otherwise I will use a document store NoSQL database such as MongoDB. This will be decided upon after the requirement gathering stage of my project as I will have a better idea of what is needed. Git will be used as a versioning mechanism of code base which will be committed to my personal and private GitHub repository. For unit tests I intend to use Mocha.js testing framework as I have had previous experience with it. I will research and try to implement a continuous deployment methodology into my project to ensure streamlined deployment process, complete with unit tests checked before deployment and syncing code to server. I hope to do a mixture of coding styles but enforcing Test driven development when necessary to ensure code clarity and maintainability. For the ability to connect external devices to the application, I will be manipulating MIDI signals from the device using the Web MIDI API as it is the most recognised way to do so.

7. Project plan

Project Plan			
Stage	**Expected Start data**	**Expected Completion date**	**Products/Deliverables/Outcomes**
1. Initiation		01/02/17	PID
2. Investigation of requirements, resources and project planning	30/01/17	06/02/17	Investigation of things necessary to get my project started. Planned and formalised starting points of my project in Trello
3. Initial code base and framework implemented and planning of high level design	07/02/17	14/02/17	A simple HTML page output with the backend padded out with correct initial and basic routing. Design documents for GUI and backend ideas
4. Increment 1. Core product bare bones implementation	15/02/17	01/03/17	A user can create a simple pattern with basic preloaded samples, real-time and user log in to portal implemented. Unit tested.
5. Increment 2. Increase functionality of user portal and musical features enhanced	02/03/17	15/03/17	The web portal has it's own page with stored information about the tracks. The musical composing tool has sound manipulation features as well as ensuring more than two can work on track
6. Increment 3. Ensure previous development is fully tested and working. Improve on those features by starting to implement desirables	16/03/17	29/03/17	Tests for core product implemented. Any other aspects improved and simple desirables started to be implemented
7. Increment 4. Implement MIDI features and other features for the musical composition tool	30/03/17	10/04/17	MIDI implemented allowing users to control music with external tool, users should be able to download music, chat feature enabled as well as desirables features (see section 4)

7.1 Control plan

1. Highlight reports as dictated by the PROC304 module
2. Review meetings with project supervisor as dictated by the PRCO304 module; additional ad-hoc meetings as and when necessary. Notes of meetings shall be documented.
3. Risk management (see Section 8); communication plan (see Section 7.2); quality plan (see Section 9); exception reports and plans as necessary
4. I will use online project tracking tool 'Trello' to plan my project. In this I have boards that have general things that are needed for the project and within further columns in which I can move cards, which are more specific tasks to the board, from left to right to track progress. I will also do this to track the work my fortnightly sprints. I will use boards in the following ways:
    a. To have board on the project initiation which is what I need to do to get the project at a level where I can start working on the code. This include cards with check-lists for setting up server, bare bones code base, project documents etc.
    b. To have a project backlog board which houses all my project ideas, requirement gathering for them, when requirement gathering is done moves to ready for story point estimation and then finally sprint candidates.
    c. To have a board that tracks my current sprint. This will have columns for Sprint backlog (list of user stories for the sprint), In progress, a column for QA-ing (making sure it hits requirements and further testing of functionality), ready for release and done (a card will move into done when it is deployed). I have an extension to Trello which allows me to assign and track user story points for each user story. This will allow me to refine my assigning which ultimately will give me a better idea for how much work I have and how long it will take. I plan to show progress in a burndown chart.

7.2 Communication plan

I have no further stakeholders or real-clients apart from myself and my supervisor. When my product is to a presentable core working prototype, I intend to hold regular meetings to get constant feedback on the product. When I seek feedback from potential users, I will plan meetings with a individuals or groups with a script and make sure the discussion of the meeting is documented fully. Review meetings will be held with the supervisor in line with the Control plan. Further ad-hoc communications will take place as needed. Unexpected outcomes or exceptions will be documented and brought to light in the review meeting unless seen as a matter of urgency.

8. Initial risk list

Initial risk list	
Risk	**Management strategy**
Schedule Overrun	Contingency has been incorporated into the project plan as well as regular meetings with my supervisor where there will be a constant monitor of the progress. Trello is put in place to keep track of individual project points with time taken on the point included in their documentation. Weekly highlight report will bring to light slippages in schedule.
Difficulty learning and using technology	A very basic core product is built into the plan from a very early stage to ensure completion. Any extra different features with different technologies will be included as a desirable if not a part of core product.
Technology failure	Git will be used to ensure code completed is always available. Regular back ups will be in place for any other project related technology such as server and documents.
Personal issues or illnesses	This is linked to schedule overrun and as such, any time taken off of project to deal with personal issues will be catered for in the contingency built into the project plan.
Requirement bloating	Also linked to schedule overrun however the core product is defined (see Section 4) and is the focus before moving onto any other requirements that may be defined.

9. Initial quality plan

Initial Quality plan	
Quality check	**Strategy**
Requirements	Requirements will be tracked, checked and monitored throughout project with aid of Trello and supervisor to ensure completion. Core versions of system will be externally tested by users, prototypes and walkthroughs.
Code implementation and functionality	I intend to have high code coverage of tests which will be implemented using Mocha.js,

	results of which will be documented and monitored. Unit tests and such will be tested on deployment to ensure a continuous delivery methodology.
User acceptance	As soon as my product has working core functionality I will continuously delivery prototype in order to ensure that user and myself is satisfied with progress and direction of product and its interface.
Documents	Documentation of development, highlight reports and other notable extensions will be checked when necessary by supervisor and uploaded to the SPMS.

10. Legal, ethical, social and/or professional issues

Music samples used in the software will be royalty free and free to use completely without copyright infringement to abide by laws. If the functionality to be able to upload own samples is implemented, a disclaimer will be in place to make sure that the user accepts full responsibility for the use of samples that are not royalty free. If the inbuilt text messaging feature is built, I will have to make sure that the user acceptance responsibility for any text communicated in the product. As the user's have to know each other to share the link one would assume that this won't be a problem, however I believe a monitoring tool inside program is outside of the scope of this project.