Museum of Sound Mobile and Web Application

System Design and Specifications

CITS5551

Members:	**Student ID:**
Bryan Trac	21704976
Jeremiah Leo	21734544
Megat Hisham	20754665
Nicole Low	21151969
Xin Wang	22167361

Table of Contents

1 Introduction

The Museum of Sound is an application that allows users to capture sounds in real life and real time, and associate them with metadata for uploading. The sounds are then curated, and made available to musicians, researchers and the general public for use in compositions, further research, education or general interest.

A web application prototype was built for this application using Django, but is in need of architectural and programming revision for actual production use. A mobile application is required to make the recording, tagging and uploading of sound a simple and user-friendly operation for users. The project client is Dr Sarah Collins, Senior Lecturer in Music at UWA.

1.1 Purpose of the document

The Functional and technical specification and project plan provides detailed information on how the software solution will function and its requested behavior. This document is created based on the requirements identified in the stakeholder meetings and provides traceability on the functional specifications back to the stakeholder requirements. Included in this document will be the detailed functional requirements including use cases, system inputs and outputs, process flows, diagrams, and mock ups.

1.2 Project Scope

The software system developed in this project will be a web and mobile application that allows users to capture sounds in real life and real time and associate them with metadata for uploading. The user will add a narrative to the sound recordings through a description, title and tags to associate an emotional meaning to the sound recording. The system will be for musicians, researchers and the general public for use in compositions, further research, education or general interest. The application will be used to better understand what sounds are meaningful and why to users, and to generate conversations about the sounds around us in everyday life. This system will be designed to promote the exploration of everyday sounds in music production, by providing tools to assist in finding and listening to uploaded sounds based on categories and descriptions. This would otherwise have to be performed manually by searching over many websites and sources. The system will meet the user’s needs while remaining easy to understand and use.

More specifically, this system is designed to allow users to record, organise and explore a collection of uploaded sounds, and download and use in music production environments. The application will facilitate recording and uploading, as well as an easily navigable explore page that allows users to view posts based on the metadata associated with each recording. The system will also contain a relational database containing a list of Sounds, Authors, Tags, Descriptions and Locations.

The project will require a redesign of the current website so that it is working, functional and user friendly. A mobile application will be developed based off of the web application. The project will involve a re-design of the database, the metadata model and a re-consideration of the use cases for this application. The web application should provide a mechanism for users to live record or upload a pre-recorded audio file of sounds. The user will additionally be able to describe, tag and title their audio files. Users can then explore audio files through the tags, descriptions and titles; and download them for creative or personal use. Each user will have an account and editable profile, and administrator control and audit facilities will also be provided.

1.3 Objectives

- To perform analysis and design of the web application by the 25th of October, 2019.
- To design a system that meets the user requirements set out by the client, by the 16th of June, 2020.
- To design a system that allows users to record, upload and explore sounds, with web and mobile applications.

1.4 Success criteria

The system is developed within the budget provided by UWA

- No budget allocated for analysis and design beyond labour

The system is developed within the resources available to the team

- Five full-time university students available for analysis and design
- Each has their own laptop

The system is developed within the timeframe set by the client

- Analysis and design work/documentation complete by 22nd of September

Realised benefits:

- A working and functional website (museumofsound.io), accessible by users and administrators
- A working and functional mobile application accessible by users and administrators
- Users are able and willing to use the system
- The system is a long-lasting solution - it will be used within UWA and externally, with updates allowed (provided no unprecedented societal or technological change)
- Reliable - tentatively set an uptime benchmark of >99%

Stakeholder satisfaction:

- The client accepts the two applications - that is, upon receiving it they are willing to use it
- Users accept the system
  - they are willing to use the system (we will tentatively set a benchmark of at least 75% of account holders using system within 12 months of introduction)
  - there is not an unacceptable level of complaints (we will tentatively set a benchmark of under 20% of users dissatisfied with system 12 months after introduction)
- Regulators (government, or industry bodies) accept the system - the system is licensed correctly for use of audio recordings

1.5 Risks and Assumptions

The web application is currently constructed using Django, a Python-based free and open-source web framework, which will continue to be used.

The users or admin will have a Web browser that supports the Django framework, with any Operating System as long as it supports an appropriate Web browser. To open the mobile application, the user will have a device that supports either iOs or Android.

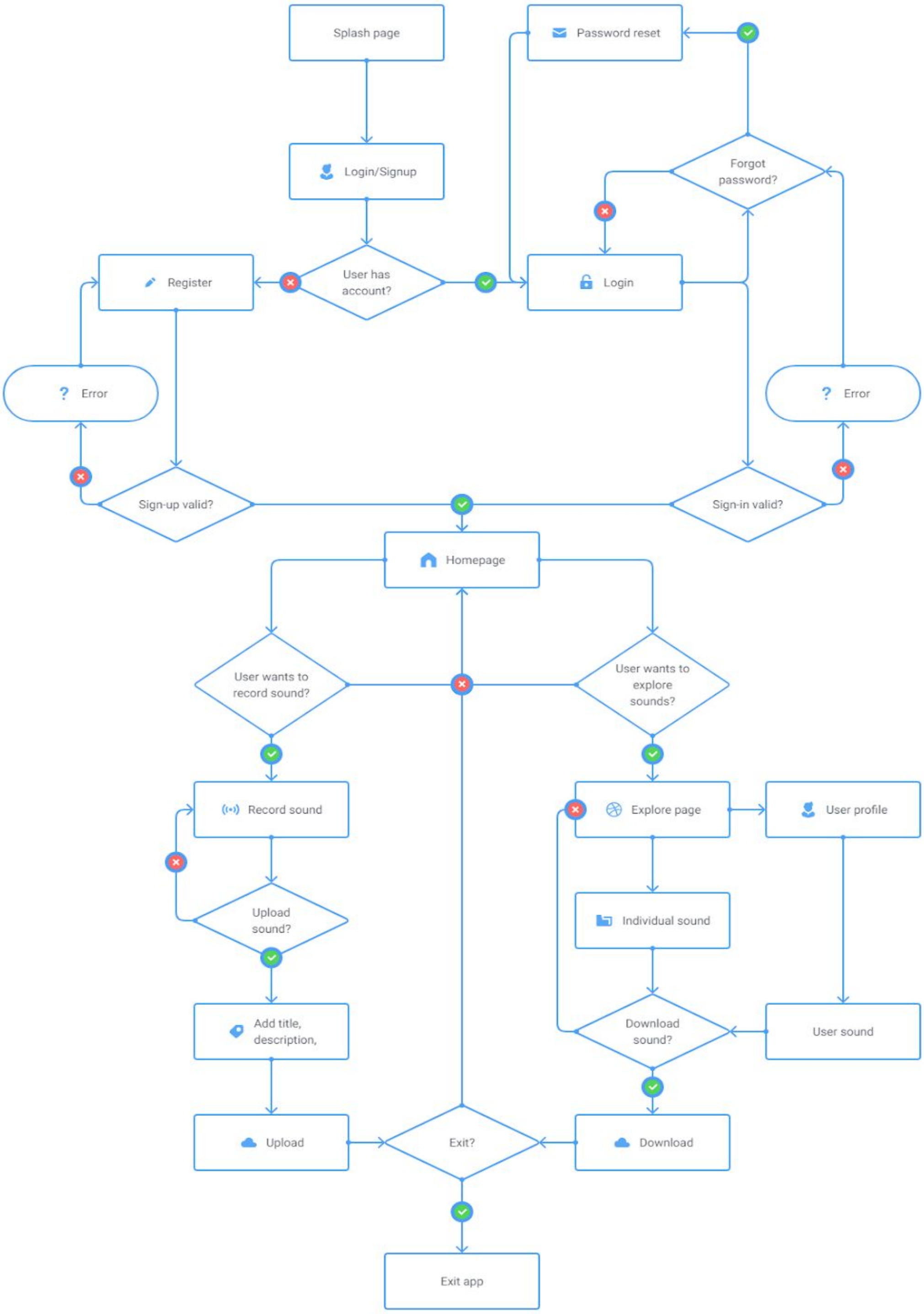
The web and mobile applications will be hosted by a third party; Amazon Web Services.

The uploaded sounds will be licensed under creative commons and be available under fair use. There will be an administrator function that allows posts to be deleted if they do not comply with this licensing.

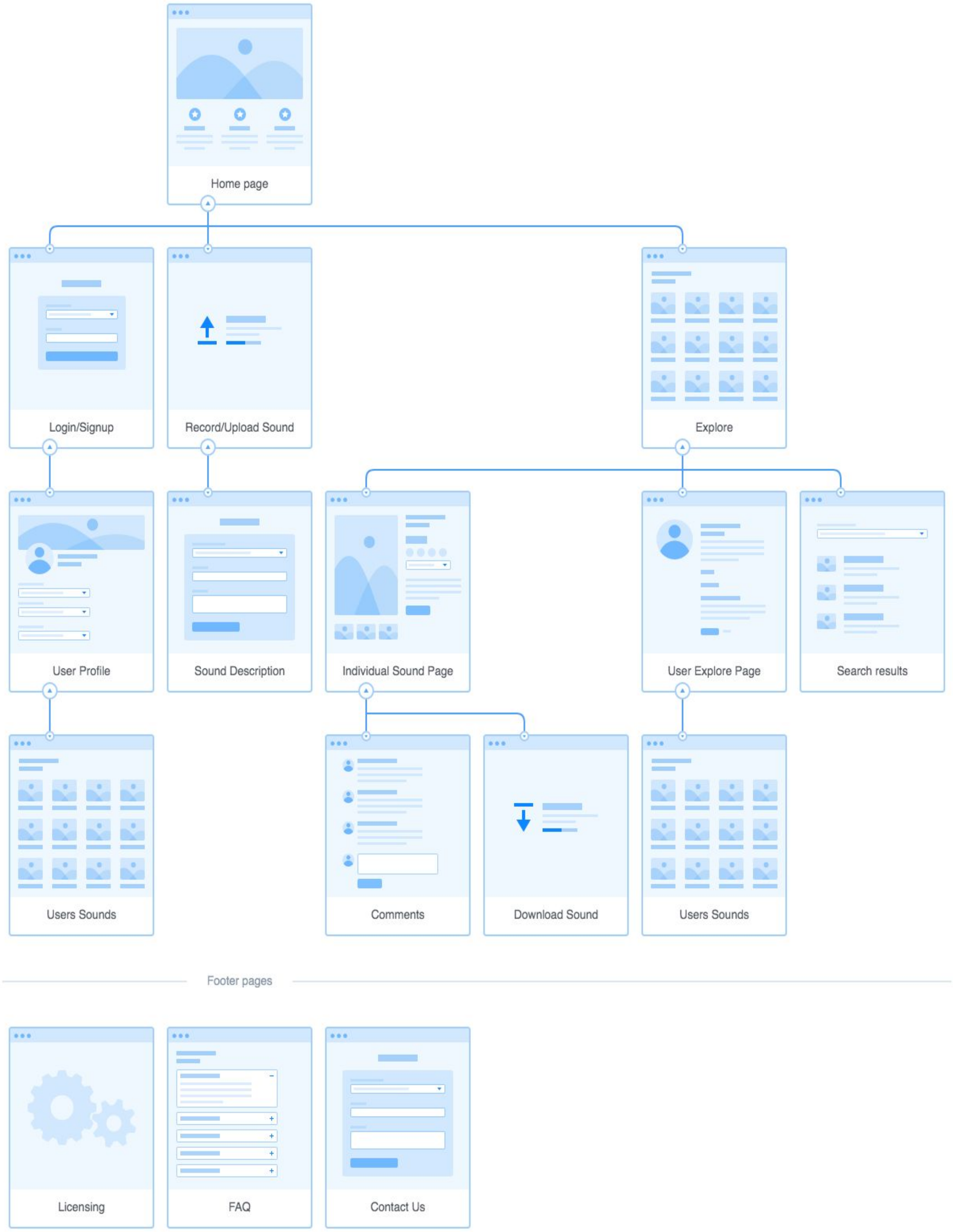
The web domain name is already owned by David Glance and the Department of Computer Science and Software Engineering, and all costs associated with this will be covered by the department.

2 System/ Solution Overview

2.1 User Flow Diagram



2.2 Sitemap, Process Flow



2.3 Scenarios/Use Cases

User record audio and uploads it

Bob walks around a Perth street, and he finds that the sound of the bustling street attracts his interest. He feels that the sounds of this street are meaningful to him as they are an audio reflection of a typical busy, Perth street where he lives. He wishes to record and upload this to the Museum of Sound. He quickly pulls out his phone and goes to the museum of sound webpage, “www.museumofsound.io” from a browser, and is redirected to the splash page. Bob is an existing user of the Museum of Sound website, and is already logged in to his account. He clicks on a big red button which says record, and his phone’s microphone begins recording. There is a limit to only three minutes of recording. After fifteen seconds of recording, Bob clicks on the red button again which stops the recording. He then decides that he would like to extend the recording, and he clicks the red button again to continue. After another two minutes and forty-five seconds of recording, it hits the three minute limit and the application immediately stops recording. Bob then clicks on an image icon, clicks the upload picture option, and chooses a picture from his gallery or take a picture with the phone camera. Bob clicks the upload button and a confirmation message appears. The file will then start to upload, a bar is displayed on screen that displays the progress of the upload, and once that bar reaches one-hundred percent a message will confirm that the audio has been successfully uploaded. Bob is then prompted to fill out a description that asks for a narrative surrounding why this sound is meaningful to him. Additionally, the page requests relevant tags for the sound and a title.

Removing audio file

The administrator goes into the Museum of Sound website, logs into their administrator account and checks a list of existing recordings. They received an email from a user regarding an unintended upload, and they browse through the audio recordings to find the specified recording stated on the email. They find that there is one audio file that is inappropriate and matches the emails description. They click on the bin icon, click confirm and remove the unintended uploaded audio.

User sign up

Charlie goes into the Museum of Sound website and decides that it is going to be beneficial to sign up and become a member. He clicks the ‘Sign Up’ icon on the top right of the splash page. He enters his name, email, password and confirm password. He reads the terms and conditions, tick the ‘accept terms’ box and clicks the sign up button. He is redirected to a page that asks him to check his email for a confirmation email. Charlie logs in into his email, and finds the confirmation email that is automatically sent. He reads the email and clicks the confirmation button. Charlie is then redirected into the website congratulatory webpage that says he has successfully registered as a member of the Museum of Sound.

User search and download audio file

Megat goes into the Museum of Sound website and logs in as a member. He is curious to see what audio files have been recently uploaded on to the Museum of Sound, so he clicks on the explore page link on the top right corner of the home page. He is redirected to the explore page where there is a list of published audio files. He types ‘birds’ in the search bar and he is redirected to the search results page. he scrolls through the list and plays each of the audio files. He likes one of the audio files, and decides to download it. He clicks the audio file title, and he is then redirected to the sound page where descriptions and comments can be read. He clicks the download icon found on the page. The audio file starts to download, and a progress bar is shown. Once the progress bar is filled, the file is fully downloaded.

Use Case ID	**UC-1**
Use Case Name	UploadingSound
Use Case Goal	To upload recorded sound
Actors	User, Web Page, Database
Linked Requirements	3.2, 3.3, 3.4, 3.5, 3.6
Flow of Events	1. A form will show for users to fill in the details such as a title, description, tag, image 2. User will click confirm 3. A message will show to check if user wants to remove or upload it the file to the website 4. User clicks upload and the audio file will be uploaded to the website
Alternative Flows	4' User clicks remove, and the audio file will be removed and not uploaded
Preconditions	User is in the webpage User has recorded an audio file
Postconditions	Audio file will be uploaded to the AWS cloud storage

Use Case ID	**UC-2**
Use Case Name	SignUp
Use Case Goal	To allow user to be registered to the website
Actors	User, Web Page, Database
Linked Requirements	3.6
Flow of Events	1. User click on the sign up icon 2. User redirected to a signup page 3. User keys in their details 4. User clicks signup 5. Email verification will be sent through 6. User verifies and is officially signed up
Alternative Flows	6' User did not verify, after twenty-four hours, the verification email expires
Preconditions	User is on the webpage
Postconditions	User is officially registered

Use Case ID	UC-3
Use Case Name	LogIn
Use Case Goal	User logs into the website
Actors	User, Web Page, Database
Linked Requirements	3.6
Flow of Events	1. User clicks on the sign in icon 2. User keys in username and password 3. User clicks sign in or taps enter key 4. User gets logged into the website
Alternative Flows	3' User did not click sign in or tap enter 4' User did not get logged into the website
Preconditions	User is in the webpage User is registered to the website
Postconditions	User is logged into the website

Use Case ID	UC-4
Use Case Name	LogOut
Use Case Goal	User logs out from website
Actors	User, Web Page
Linked Requirements	3.6
Flow of Events	1. User clicks the sign out icon 2. A confirmation message appears 3. User clicks confirm 4. User is logged out
Alternative Flows	3' User clicks cancel 4' User is still logged in
Preconditions	User is logged in User is in the web page
Postconditions	User is logged out

Use Case ID	UC-5
Use Case Name	AudioFileConfirmation
Use Case Goal	Admin to confirm that this sound file is to be displayed on the website
Actors	Admin, Web Page, Database
Linked	3.6

Requirements	
Flow of Events	1. A to be confirmed list of audio files uploaded will be displayed 2. Admin will click the confirm icon to confirm and the audio file will be removed from the list 3. Audio file will be public and users able to search and download
Alternative Flows	2' Admin chooses to not confirm, and audio will still be in the to be confirmed list 3' Audio file will not be available for the public to search and download
Preconditions	Admin is logged in Admin is the web page
Postconditions	Audio file is ready for publishing

Use Case ID	**UC-6**
Use Case Name	RemoveAudioFile
Use Case Goal	Admin removes unwanted audio file
Actors	Admin, Web Page, Database
Linked Requirements	3.6
Flow of Events	1. Admin clicks on to the confirmed audio file page 2. Admin searches for the specific audio file 3. Admin clicks on the remove icon 4. Confirmation message will appear 5. Admin clicks confirm 6. Audio file is removed from the database
Alternative Flows	5' Admin did not click confirm 6' Audio file is not removed from the database
Preconditions	Admin is logged in Admin is in the web page
Postconditions	Audio file is removed and will no longer for the public to use

Use Case ID	**UC-7**
Use Case Name	DownloadAudioClip
Use Case Goal	User download audio file
Actors	User, Database, Web Page
Linked Requirements	3.6, 3.8

Flow of Events	1. User clicks the explore page 2. User scrolls down or types in keywords to be searched 3. User clicks on the audio file they want 4. User clicks download 5. A confirmation window pops out 6. User clicks confirm 7. The audio file will start downloading 8. Download finished
Alternative Flows	6' User clicks cancel 7' The audio file will not start downloading
Preconditions	User is logged in User is on the web page
Postconditions	Audio file is downloaded into the Users device

Use Case ID	**UC-8**
Use Case Name	SearchAudioClip
Use Case Goal	User searches the database for audio file that contains keyword
Actors	User, Database, Web Page
Linked Requirements	3.7
Flow of Events	1. User clicks the explore page icon 2. User can either scroll down or type in keywords into the search bar 3. Audio files with the keywords will be displayed
Alternative Flows	-
Preconditions	User is on the web page
Postconditions	Audio files with the given keyword is displayed for the user

Use Case ID	**UC-9**
Use Case Name	RecordAudio
Use Case Goal	User to record audio
Actors	User, Web page
Linked Requirements	3.1, 3.6
Flow of Events	1. User is on the home page 2. User clicks the red "Record" button 3. Application will start recording sound 4. User clicks the button again 5. Sound will stop recording 6. Audio file is created

Alternative Flows	4' Audio records for 3 minutes 5' Audio immediately stops recording 6' Audio file is created
Preconditions	User is on the web page User is logged in
Postconditions	Audio file is recorded and created, further action needed

Use Case ID	**UC-10**
Use Case Name	UnpublishAudioFile
Use Case Goal	To unpublish audio file
Actors	Admin, Web page, Database
Linked Requirements	3.6
Flow of Events	1. Admin clicks on the confirmed and published page 2. Admin searches for the specific audio file 3. Admin clicks on the unpublish icon 4. Confirmation message will appear 5. Admin clicks confirm 6. Audio file is unpublished on the web page
Alternative Flows	5' Admin clicks cancel 6' Audio is still published on the web page
Preconditions	Admin is logged in Admin is on the web page
Postconditions	Audio file is unpublished but not deleted from the webpage

3 Functional Requirements

3.1 Record Audio

The system's main feature will be to allow users to record live sound. On both the website and mobile application, on the splash and home page, users can press a button to start recording sounds. The recorded sounds will be taken from the device's microphone. Audio clips uploaded this way will be saved as .mp3 files.

3.2 Upload Audio

Besides recording sound clips, users should also have the option of uploading pre-existing audio files to the system from their device. They may either choose to navigate through their files and select a file to upload, or drag and drop the desired file. The uploaded audio file must be one of the following file formats:

- .mp3
- .wav
- .aac
- .flac

Otherwise, the file will be rejected. If the file is one of the four, the user is permitted to upload it to the system.

3.3 Adding Metadata

Users will be allowed to add extra information to their posts in order to give them narratives. This will include:

- Title
- Description
- Image
- Tags

These will be displayed as empty textboxes which the user can fill in (besides the image). The only compulsory field will be the title. Titles do not have to be unique.

The maximum length of the title will be 50 characters and the maximum length of the description field will be 1000 characters. Adding images and tags will be elaborated upon in 3.4 and 3.5 respectively.

3.4 Adding Images

To give their posts more narrative, users should be allowed to upload an accompanying image. Users can upload an image from their computer or phone by selecting a valid image file. Drop and drop will not be enabled for this image.

When the image is uploaded, it will be scaled down to fit within a specified size, either 128x128px for thumbnail size or 512x512px, while preserving the aspect ratio. The system will support image sizes up to 100mb.

3.5 Tagging Audio Files

When uploading an image, users can add tags to the image to allow other users to find the audio recording easier (see 3.7). The tagging will be in the form of a scrollable checklist. The user can check the tags that they want from that list. There will also be a search bar at the top of the list that the user can look for the tags that they want.Users may only tag their audio files from this list of tags. The upper limit of tags per audio file will be 5.

The amount of tags will be fixed. The list of valid tags will be stored in the database and will only be able to be updated by an admin.

3.6 Geotagging

When a user records an audio clip, they may also add in the location of where the sound was recorded. This geotagging feature will allow users the clip with their current location. The location will be based off of where the user is during upload time so if the user moves after recording the sound and before uploading, the geo tag will be the location where they uploaded the file.

This option will be set to off as a default and there will be an option for the user to opt in if they want to geo tag their audio clip. The user can see the geotag before uploading the file and can choose to opt out if they change their mind. The tag will be displayed and stored as a single string.

3.7 User Account

Users will be able to make their own accounts with their own login credentials. When first visiting the page, a user may choose to sign up. Users must enter a unique username as well as a password with at least one uppercase letter, one lowercase letter and one number. When logged in users can:

- Upload under their own name
- View all their previously uploaded files
- Delete their own previously uploaded files

3.8 Search for Audio Clip

The system will feature a search function which will allow users to look for specific audio clips. Users can either search by title or search for audio clips with specific tags. When searching by title, users will fill in a search bar. When searching by tag, the user should be able to see the list of tags that exist in the system. Selecting one of them will fill the search results with files with that tag.

When the search results are displayed on screen, the default sorting order will be by upload date, where the most recent uploaded file will be at the top. Users can change this sort to one of the following:

- Upload date
- Title
- Username

The sort order can either by ascending or descending.

3.9 Download Audio Clip

When viewing an audio clip, besides the metadata, there will be a button on the page which will let users download the audio clip. Pressing the button will bring up a flyout which will ask the user where they want to save the audio file. They may either select a destination or cancel. If they go ahead, the audio clip will be saved to their computer with the title of the audio clip being the default title.

3.10 Cross Platform Support

The website system will be usable in all major browsers including;

- Internet Explorer
- Chrome
- Firefox
- Safari

The mobile application will be able to be installed and run on both Android and iOS. All the functional requirements above will be available on both website and mobile systems.

4 Technical Requirements

4.1 Usability

The existing web application seems clunky and often confusing, and user experience is not as smooth and logical as it needs to be. User interface needs to be more intuitive and easy, which will be achieved using different frameworks and a rework of the overall sitemap. Bootstrap will be employed as a front-end framework to have uniform design elements.

4.2 Platform and Frameworks

The web application will be built using Django, a Python-based free, open-source web framework. The application will follow a model-template-view architecture pattern. The web application will employ a serverless, cloud architecture, and will be hosted by Amazon Web Services (EC2). The user interface will be developed using HTML5, CSS3 and JavaScript and the application will be supported by any browsers which support these. The application will use components from standard and commonly accepted libraries such as jQuery, React.js and Bootstrap. The application will store data using AWS S3, in a non-relational, noSQL database. The application code will use Git version control, and all commits will be archived in a GitHub repository which is available to all team developers. Source code will be stored on a mutually agreed platform. The web application will be converted to a mobile application for iOS and Android using React Native.

There are several platforms that we need to consider within each layer when designing the web and mobile application. The layers of concern are the data layer, the structure layer, design layer, behavioural layer and the application layer[5].

- Data layer: AWS S3 (Computer cloud capacity C2 to host website)
  - We do not have a big enough storage system to facilitate the web site or locally store data

    - Non-relational/No-SQL data storage
    - Serverless architecture**
- Structural layer[5]: HTML
    - Most common markup language
    - Covers all types of browsers
- Design layer [5]: CSS
    - Using stylesheets (bootstrap)
    - Most popular front-end framework
- Behavioural layer [5]: JavaScript
    - Will be using Ajax to have an asynchronous recording and uploading application
    - React.js
- Application layer: Django (Python)
    - Previous developers used Django and keeping this will save migration to another framework

In terms of deployment we will be using the four-tier deployment model. This entails [2]:

- Development:
    - Where the developer makes changes to the code and is usually a local, single tenant environment (i.e. personal laptop) [2]
- Testing:
    - An integration environment where developers merge changes to test that they work together [2]. System acceptance tests.
- Staging:
    - Running the tested changes against an existing web application that has the same services as we do (production-equivalent infrastructure and data to ensure they will work properly when released) [2].
- Production:
    - Run the museum of sound web application in a live production environment

The operational considerations that we need to consider when developing the web application would be:

- The cost and difficulty of acquiring and managing computing resources within AWS, as not all of our members have previous experience with it.
- The challenges in code merging within the development tier.
- The challenge of some changes can’t be easily tested in isolation, this causes locating and verifying those changes harder.

4.3 Performance

The system has various features that work and interact together. All features should be responsive and fulfill their intended purpose in a reasonable amount of time. The software will need to adhere to the following performance criteria.

- Users should be able to navigate between various pages almost instantaneously (within 1 second).
- When uploading audio clips, the uplaods should be fast and take no longer than 10% of the length of the clip itself. The cap on the max length of an audio clip is [three minutes, subject to change] so the maximum upload time should be 18 seconds.
- The mobile app should have similar wait and response times as the website.

For the search feature, the more audio clips in the database, the more items that will need to be searched. As the database gets bigger, the search feature will take longer to return matches. For this project, the planned maximum user base for the prototype is 20 users. If 20 users are able to simultaneously use the system without any noticeable decrease in performance, it will be considered a success. However, both the app and website should be scalable in such a way that in the future, many more users can be accommodated at the same time. To ensure the system is scalable

- Careful thought will need to be put into the way the database is structured
- How items are searched for in the database

4.4 Security

Web application security is the process of protecting the application malicious attack that exploit the applications vulnerabilities in the applications code. Motivations as to why web applications are considered as a very attractive targets for malicious attacks are simply because due to the inherent complexity of the applications source code, there is a high chance of some unattended vulnerabilities that the hacker could exploit. The ease of executing this attack also plays a big role in incentivising the attacks as it could be easily automated and launched to a vast number of applications at a time[4].

There are several aspects of AWS S3 that we will be focusing in terms of security implementation. These are:

- Authentication. These are confirming that the communicating entity is the one claimed
- Access control. This is preventing unauthorised use of resources
- Data confidentiality. Protection of data from unauthorised disclosure
- Data integrity. This is the assurance that the data received is sent by an authorised entity.

- Non-repudiation. Which is protecting against denial by one of the parties in a communication
- Availability. This is managing the resources that are accessible or usable.

We will be managing this by assigning permissions through the AWS. With regards to the web applications code design it self, we would incorporate the following sets of securities:

- Password encryption [1]. This could be password hashing.
- Moderating error messages [1]. This is in order to make sure that not all information are given away during an error event
- Utilizing https [1]. This is a provide security over the internet in that it guarantees that the users are communicating to the correct server.
- Validations of inputs from user on both browser and server side [1]. This is to avoid malicious code being inserted into the database(partially covered within the AWS S3 permissions management)

4.5 Maintenance

Maintenance is an important part of keeping you users happy and retaining them. As minor problems can cause unnecessary irritations and descentivise them to use the application, minor improvements enhances the users experience and incentivise them to use the application more. This shows to them that the owner of the application cares about their feedback. Our plan is to set a regular maintenance in the future after the product launch. As with a regular maintenance plan the cost would be much less as it is manageable and predictable, compared to having a massive update when the product critically fails[3].

There are a number of possible maintenance that we would like to expand on within this web application. These are:

- General monitoring and bug fixes [3]. This is daily check up for any unexpected software errors hindering the users and fixing it.
- Security patches and updates [3]
- Implementing new functionality in future [3]. This could be adding location of recording through google maps, real time chat using facebook messenger.
- Third-party API update [3]. An example of this could be the potential google map's map updates for pinpointing where the sound was recorded
- Application scaling [3]. This is situations where as the application gets more and more traffic/users an increase of the AWS server capacity and resources is required.

4.6 Globalisation and Localisation

The main file formats that the database will be dealing with are audio files. These will be stored as mp3 files. Mp3 files are a universally used file format that every operating system and region uses so no extra consideration is required for file conversion. The plan is for the app to be mainly used in English speaking countries so there currently is not any plans of branching off into other languages. However, if this were to become a desired feature in the future, the system should be built in such a way that it is able to easily accommodate for this. It will need to have simple clear headings and intuitive navigation that is able to be translated across multiple languages.

5 Out-of-Scope Requirements

5.1 Geotagging Using a Map or Search Box

Geotags allow users to enhance the stories they tell with the sounds that they upload. However, users will only be able to add geotags by letting the system know their current location. Having an option which lets users search and input locations using a map or search box will allow for more flexibility but it will introduce more complexity to the geotagging feature. It also introduces possibilities of incorrect geotags. So to keep the system a bit simpler, the geotagging will be limited to using current location.

5.2 QR Code Scanning

This scanner will allow the users to quickly and easily upload audio clips by scanning QR codes that they find. It can improve the usability and convenience the system but is not a necessary component of the system. This feature will not be implemented in order to prioritise more important functional requirements.

5.3 Advanced Search

When searching for audio files, users can either search by title or by tag. Additional sophisticated searching options such as filtering upload date, seen often in other applications, will not be implemented. As this is the first instance of the search feature being implemented for this system, more focus will be placed on ensuring it is correct and optimised before extra features are considered.

5.4 Order Search Results by Popularity

One of the most popular ways to order search results is by the number of hits. Implementing this will require extra frameworks. The amount of time that will be needed to be allocated to this small quality of life feature is fairly significant. Therefore, this feature will not be present in the system.

5.5 Real Time Chat

The creators of the current prototype system attempted to implement a real time chat using Facebook’s API. However, due to issues with permissions, this feature could not be implemented. As the feature is difficult to implement and is not integral to the main functionality of the system, time will not be spent on it.

6 Restrictions and Constraints

6.1 Timeline

The project has a strict inflexible project delivery date. As such, as the delivery date approaches, scope changes will be extremely difficult to accommodate for as the project cannot be given any extra time. There are also strict dates for deliverables that have to be met so the project timeline will need to be structured around them.

6.2 Budget

The budget of the project is not too much of a concern as a sufficient amount of money can be allocated as needed and profit is not a consideration.

6.3 Resources

The project team consists of 5 members all of which have various other commitments. The amount of labour hours that can be allocated to the project is heavily limited. If any important changes to requirements occur, especially the addition of new functional requirements, labour hours will need to be reallocated from another requirement. As a result, this old requirement will either need to be cut down or excluded entirely.

7 References

1. Gerber, R. and Perry, T. (2019). *9 security tips to protect your website from hackers*. [online] Creative Bloq. Available at: https://www.creativebloq.com/web-design/website-security-tips-protect-your-site-7122853 [Accessed 22 Sep. 2019].
2. Platform.sh. (2019). *What is best practice in web application deployment?*. [online] Available at: https://platform.sh/blog/what-is-best-practice-in-web-application-deployment/ [Accessed 22 Sep. 2019].
3. Rubygarage.org. (2019). *6 Categories of Web Application Maintenance: From Bugs to Scaling*. [online] Available at: https://rubygarage.org/blog/web-application-maintenance [Accessed 22 Sep. 2019].
4. Learning Center. (2019). *Web Application Security* | *Application Security Checklist* | *Imperva*. [online] Available at: https://www.imperva.com/learn/application-security/application-security/ [Accessed 22 Sep. 2019].
5. Stenström, E. (2019). *Concept: Four layers of web development*. [online] Friendly Bit - Web development blog. Available at: https://friendlybit.com/css/concept-four-tier-web-development/ [Accessed 22 Sep. 2019].

8 Appendix

Project Plan

Project development schedule

Stage	**Task**	**Deliverables**	**Planned start time**	**Planned termination time**	**Responsible**
System requirements	Research user needs and user environment		14 Aug-19	22 Sep-19	All members
	Develop a preliminary plan for the project		14 Aug-19	21 Aug-19	Xin Wang
	Development plan review	Project Plan	14 Aug-19	22 Aug-19	All members
Analysis	Determine the system operating environment		23 Aug-19	29 Aug-19	All members
	Establish a system logic model	Project development plan	23 Aug-19	29 Aug-19	Jeremiah Leo, Nicole Low
	Write requirements specifications	Functional and technical specifications document	30 Aug-19	19 Sep-19	Functional- Nicole Low, Jeremiah Leo Technical- Bryan Trac, Megat MegatHisham, Xin Wang
	Confirm the project development plan		22 Sep- 19	6 Oct-19	Sarah Collins, David Glance
Summary design	Establish the overall structure of the web app and divide into modules	Wireframes, mockups	6 Oct-19	20 Oct-19	All members
	Define each module interface	Application design specification	6 Oct-19	20 Oct-19	All members
	Database design/ AWS integration	Database design specification	6 Oct-19	20 Oct-19	Bryan Trac, Xin Wang
Detailed design	Define coding requirements for each module	Pseudo-code	24 Feb-20	1 April-20	All members
	Determine the detailed interface between modules	Pseudo-code	24 Feb-20	1 April-20	All members
	Develop a modular test plan		24 Feb-20	20 April-20	Nicole Low, Megat MegatHisham
Implementation	Write the web applications source code		24 Feb-20	18 April-20	All members
	Perform module testing and debugging		24 Feb-20	20 April-20	All members
	Convert web application to mobile application		24 Feb-20	20 April-20	Jeremiah Leo, Bryan Trac

	Write user manual	User manual	1 April-20	18 April-20	
	Review the implementation process and completed documentation		1 April-20	18 April-20	All members
Acceptance Test	Test the entire web application (robustness test)	Test report	20 April-20	30 April-20	Nicole Low, Bryan Trac, Xin Wang
	Test entire mobile applications (robustness test)	Test report	20 April-20	30 April-20	Jeremiah Leo, Megat MegatHisham
	Trial user manual	User manual	1 May- 20	21 May-20	All members
	Write a summary report	Development work summary	1 May-20	22 May-20	All members
Deployment	Deploy mobile and web applications		1 May-20	22 May-20	All members

Contingency

While strictly implementing the task schedule, a weekly face-to-face meeting plan is developed to ensure that each team member completes the assigned tasks within the specified time. There are some possible contingencies, which include:

1. Encountered an unsolvable problem at work. While it is possible to slow down the progress of the task, other team members should be notified to explain the situation and ask for help.
2. The tasks are interrupted by hardware damage, system crash, data loss, etc. All members need to upload completed work to the cloud server in time so that the work progress can be quickly restored to ensure smooth work and data security. A GitHub repository will also be used for versioning and code backup.