60 Seconds

# Requirements Specification

**REVISION HISTORY**

| Revision # | Author | Revision Date | Comments |
| --- | --- | --- | --- |
| 0.1 | Full Team | February 2, 2022 | 1st Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

System Overview …………………………………………………………………………………... 4

System Block Diagram ……………………………………………………………………………. 5

Document Overview …...…………………………………………………………………. ………..6

General Requirements …………………………………………………………………………….. 6

Features and Requirements....……………………………………………………………………. 6

Use Cases ……………...………………………..……..…….…………………………………….. 7

Resources ……………...………………………..……..…….…………………………………….9

## System Overview

60 Seconds is a ***web-based application*** designed to ***record short, daily, audio recording segments***. It will be compatible with desktop/laptop computers via a web browser. Daily, a user will ***record audio*** segments ***up to 60 seconds*** in length. Their daily recording time is ‘pooled’ similar to ‘swipes’ on Tinder. Once the user ***exceeds their daily limit of 60 seconds***, they will be ***unable to record more content***. A user ***WILL*** be able to record ***multiple segments daily***, so long ***total recording time does not exceed 60 seconds.***

For example, in a single day, Johnny records 10 seconds of audio, and 20 seconds of audio; Johnny recorded 30 seconds of audio. If Johnny records any more audio that day, his ***recording CANNOT exceed 30 seconds***, or his ***limit will have been reached*** for that day. Once a limit is reached, the user must ***wait 24 hours to record more content***. The audio recordings that a user makes ***MUST be saved*** for the ***current day***. Recordings cannot be made for days in the past or the future. For example, if today is Wednesday, any recordings Johnny makes will all be stored under Wednesday.

Once audio segments are recorded, they’ll be ***available for the user to review prior to uploading*** (similar to Snapchat’s review of photo/video before sending). Once a recording is made, the application will ***automatically include default tags*** that will be ***stored with the recording***. These tags will be ***produced based on the recording’s metadata*** *(location, date, time, length.)* While the user reviews their audio segment, a description of the recording can be included. Additionally, the application will ***produce a speech-to-text transcription*** of the recording’s contents, which can be edited. When the user uploads the audio recording, this data will be stored alongside it in the database.

Implementation of this project will involve ***React (a JavaScript based front end framework), Node (a back end JavaScript runtime environment) and Amazon AWS (an API hosting service.)*** The most unique feature of this application will be the ***60 seconds Calendar interface***, an interactive in-app calendar where the user’s daily recordings will be housed, acting as the ***account ‘dashboard’***. This in-app calendar will appear on the screen, and the user will have the ability to interact with it.

* The ***current month is displayed*** after logging in to the application
* The user can ***navigate through months*** by clicking the left/right arrow keys
* Individual days can be accessed, displaying that day’s information
  + The audio segment(s) recorded that day by the user, if a recording was made
    - Playback will be available for each recording that day
    - A speech-to-text transcription display of the contents of the audio
    - Default (and custom) tags, a description, and any other information

For organizational purposes, the application will ***produce default tags based on the audio recording’s metadata*** (location, date, time, length). Users will have the option to ***add custom tags*** to their recordings with ***user-defined keywords***. Finally, a ***search bar*** will allow users to ***search for recordings*** based on their created tags, providing ***easy access to user-defined*** recording labels. Audio recordings will be available for download as an MP3 file

## System Block Diagram



Figure 1: A simple system block diagram of the application as a whole

**Figure 1** represents 60 second’s ***system block diagram***. Users with an active internet connection access 60 seconds via a client system (laptop/desktop). The web application will ***communicate with Amazon AWS’s*** various hosted services, including storage solutions, database services, and ***Natural Language Processing (speech-to-text)*** transcription services.

1. **Clients:** front-end part of the application. Will be developed using React, and will act as the application’s user interface. The user will record audio, edit the postings, and save them to the database using default (optional custom) tagging.
2. **Storage:** user information will be stored using remote storage via the cloud. Information will be called from storage using various commands. Each day a user records/uploads audio, that recording, along with any tagging, the transcription of the text-to-speech, and any other data, will all be stored here.
3. **API:** Amazon AWS will serve any and all API calls, mainly the transcription service, which will take the user audio segment, and return a transcription of the audio content. This transcription returns as a json file, which can be converted to a string to be saved to storage.

## Document Overview

The General Requirements section provides the features and requirements for 60 Seconds. It is used to define the basis for the different various testing phases:

* System Overview
* General Requirements
* Features and Requirements
* Use Cases/User Stories

## General Requirements

* Consistent network connection
* Laptop, Desktop
  + Laptop or Desktop: a modern web browser to allow access to website
    - Google Chrome
    - Safari
    - Mozilla Firefox
    - Brave
* Microphone, with proper updated microphone drivers installed

**Features and Requirements**

* Account Management
  + User registration
  + User login, utilizing Google’s Google Sign-in tool for security and ease of access
  + Local user profile management and customization
* Content Storage and Delivery
  + Store audio recordings, website content, user content in a scaling Amazon AWS cloud storage solution
  + Index each audio recording within a database for easy access and organization
  + Exporting audio for users in an MP3 format
* Content Tagging
  + Allowing users add tags to custom content they upload
  + Allowing users to search and parse through their recordings based on tags
* Audio Recording
  + Record and store audio for user to playback
  + Requires access to user device’s microphone
* User Interface
  + Interactive Calendar, with each day representing a ‘cell’
  + Each cell will display audio recording information (tags, transcript, description)
  + The current day will be highlighted
  + User can access previous recordings using the arrow keys to navigate between months (left/right)
  + Days without audio recordings will display no data, as no data was stored in the database for that cell
* Audio Analysis
  + Audio recording will be analyzed via API call to Amazon AWS Transcribe, returning a file to be ‘stringified’
  + Users will have the option to edit this string, if the transcription results do not match user input
* Secure Coding Practices
  + Secure login, HTTPS, Password protection, content management.

## Use Cases

Sarah wants to create a new 60 seconds account on her PC/Laptop

1. She opens her computer and connects to the internet
2. She navigates to her primary web browser
3. She visits 60Seconds.io and clicks on the Create Account link
4. This navigates her browser to the ‘Create Account’ screen
   1. She enters her information (email, full name, username, etc.)
   2. She creates a password
5. She clicks ‘Create’ at the bottom of the screen, and her new account is created
6. The application/website will send her a verification email to the email she provided
   1. Email will contain a link
   2. She clicks the link, her account is now ‘verified’
7. The link brings her to her newly created profile dashboard

Max wants to login to his 60 seconds account on his Phone

1. He pulls out his phone, which is connected to the internet
2. He navigates to and opens the 60 seconds mobile app
3. This opens the mobile app, which brings him to login screen
   1. He clicks ‘Login’ or
4. He is presented with a login screen and enters his username and password
5. He clicks “submit”
6. He is now logged into the 60 seconds application.

John has just created a new account and wants to customize it on his PC/Laptop

1. He visits 60Seconds.io and clicks on ‘Log-In’
   1. He is prompted to enter his account information (email/username, password)
2. Upon clicking ‘Log-In’ John navigates to his profile dashboard
3. Under his Profile Picture, he clicks the link titled ‘Edit Profile’
4. The browser navigates John to the ‘Edit Profile’ page
5. On the page, all his profile information will be displayed with various types of editors
   1. In the ‘Bio Section’ John adds a short bio about himself in the text box
   2. In the ‘Profile Picture Section’ John adds a profile picture (Jpeg, PNG, etc.) that is displayed on his profile
   3. John clicks ‘Save’ at the bottom of the screen
6. The changes made to his account are sent to the database
7. John’s profile is now updated with the new information provided

James wants to record his daily 60 second journal entry.

1. He opens his computer and connects it to the internet.
2. He visits 60Seconds.io and signs into his user account.
3. He navigates to the recording screen, and hits record.
4. He is presented with a playback of the recording when he is finished, and he can either choose to delete it and re-record or to save the current recording.

Tom wants to download his audio recording to his local storage.

1. He opens his computer or mobile device and connects to the internet.
2. He signs in his user account on 60 Seconds application.
3. He navigates to the recording history page and selects a date.
4. He touches the download icon on the date he selected.
5. The recordings he downloaded will be stored on his PC/Mobile device as MP3, and he can access these MP3 files without internet connection.

### **Resources**

* <https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/#overview>
* <https://developers.google.com/identity/sign-in/web/sign-in>
* <https://azure.microsoft.com/en-us/services/storage/blobs/>