# **Group 4 Capstone Use Case Scenarios**

# **Create Account**

## **Context**

None

## **Primary Flow**

1. Note-Taker(User) selects to create account

2. Note-Taker enters username and password

3. Note-Taker selects to create account with credentials

4. System verifies username is not in use

5. System adds account credentials

6. System shows home page with empty list of uploaded sources

## **Alternative Flow(s)**

### **Username already in use**

A4. System identifies username is already in use

A5. System prompts collector to select a new username

A6. Return to step 2

# **Login**

## **Context**

None

## **Primary Flow**

1. Note-Taker enters username and password

2. Note-Taker submits credentials

3. System successfully verifies credentials

4. System shows home page with empty list of uploaded sources

## **Alternative Flow(s)**

### **Invalid Credentials**

A3. System fails to verify credentials

A4. System notifies Note-Taker and prompts to try again

A5. Return to step 1

# **Upload a Source**

## **Context**

Note-Taker is logged in

## **Primary Flow**

1. Note-Taker selects to add a new source
2. Note-Taker selects the source type, from a list of acceptable types, that they want to upload
3. System prompts Note-Taker with the correct input fields or file-upload button that corresponds to the source type.
4. Note-Taker inputs or selects the source they want to upload
5. Note-Taker selects to upload the source
6. Note-Taker adds tags related to the source
7. System stores the source metadata and the user-info into the database
8. System redirects the user to the media source page.

# **Add notes to source**

## **Context**

Note-Taker is logged-in and source file is selected

## **Primary Flow**

1. Note-Taker selects “Add Note” button when viewing a source
2. Note-Taker types notes to a specific timestamp (Only if it is a video)
3. Note-Taker selects save notes (May implement auto-save feature)
4. System inserts a new row into the notes database table that links the notes to the source file and user information.
5. System displays a confirmation dialog that a note has been saved.

# **Updates an existing note**

## **Context**

Note-Taker is logged-in, source file and note is selected

## **Primary Flow**

1. Note-taker selects the edit button
2. Note-taker modifies the note
3. Note-taker saves the update
4. System updates the note content in the database
5. System displays a confirmation dialog that a note has been saved.

## **Alternative Flow(s)**

### **User selects Cancel**

A2. Note-taker selects “Cancel”

A3. Note content returns to previous saved state

# **Removes a single note from a source**

## **Context**

Note-Taker is logged-in, source file and note is selected

## **Primary Flow**

1. Note-Taker selects remove button
2. Systems displays Confirmation dialog if the Note-taker is sure they want to remove the note
3. Note-taker selects “Yes”
4. Systems removes not from the application and from the database
5. Dialog is removed from the screen and returns back to the source page.

## **Alternative Flow(s)**

### **User selects No**

A3. Note-taker selects “No”

A4. Return to step 5