Gratitude++ Requirements Specification

By: Joel Scott, Benjamin Seifried

TABLE OF CONTENTS

PROJECT NAME

The team has decided to create a web application journal that helps focus positive events and feelings towards a sense of gratitude for the user. Taking this into consideration as well as this is a computer science project, we have named the project Gratitude++.

TEAM MEMBER NAMES

See Figure 1 in the appendix for detailed description

ABSTRACT

This project proposal outlines the development of a dynamic web application designed to empower users into cultivating a sense of gratitude through personalized journaling. The application will be built using a technology stack including React, TypeScript, Bootstrap and Firebase.

The application will allow a user to express themselves using various content formats including text, video, and audio files. The daily titles of the journal entries will be utilized in creating an interactive “word-cloud” that will display keywords more prominently depending on their number of inclusions. By blending technology and personal expression, the application aims to promote mental well-being and foster an “attitude of gratitude" and provide users with an innovative and meaningful experience.

TOOLS AND TECHNOLOGIES

See Figure 1 in the appendix for detailed description

REQUIREMENTS LIST

1. Sign-in Form:

1.1.: (Onboarding) Users will be given the option to sign in through google services

1.1.1: On click, users will be redirected to another page to complete sign in procedure

1.1.2: Users will be prompted to enter an email address and password to complete sign in

1.1.3: Authentication for sign in security will be handled through Firebase authentication services

1.1.4: Sign in flow for users will be handled manually and passed to Firebase for authentication

1.1.5: If sign in fails, users will be prompted to try again

1.2: Upon successful sign-in a user will either be directed to

1.2.1. User’s home page (if account is present)

1.2.2. Registration Modal (No account Present)
(see Registration Modal)

2. Dialog boxes (General)

2.1: Every dialog box will have a submit button and a cancel button

2.1.1: The ok button will confirm the user's selection to complete the action and close the dialog box

2.1.2: The cancel button will close the dialog box

3. User account registration modal (Onboarding)

3.1.: Users will be required to create an account in pop up modal

3.1.1.: Accounts must have an account name

3.1.2: Account names will be required to meet certain criteria

3.1.2.1: Names cannot begin with a number

3.1.2.2: Names cannot contain any spaces

3.1.2.3: Names must contain more than one character

3.1.3: All account usernames must be unique

3.1.4: Names will be confirmed by clicking the "confirm" button

3.1.4.1: On click, checks for existing name in the database

3.1.4.1.1: Sends a request to the database to check for a matching username, returns true or false

3.1.4.1.2: If no match is found, confirm the user's selection and send the username to the database

3.1.4.1.3: If a match is found, prompt the user to try again and repeat 3.1.2 - 3.1.4)

3.1.2.: Accounts must have an account avatar

3.1.2.1: The users will be given a choice of preselected avatars to choose for their account

3.1.2.2.: If the user does not choose an avatar, they will be given a default image

3.1.2.3: All avatar selections will be stored in the database along with the user's username

3.2.: Upon successful completion of the sign in the user will be directed to their personal home page

4. Page Navigation

4.1: There will be a navigation menu to navigate the site

4.2: There will be a place to display the username and avatar

4.3: There will be a tab called "My Homepage"

4.3.1: This tab will take the user to their homepage

4.3.1.1: This will display user information, including:

4.3.1.1.1 Avatar

4.3.1.1.2 Username

4.3.1.1.3 Journal Entry Component

4.3.1.1.4 Word Cloud Component

4.3.1.1.5 Delete User Button

4.3.1.1.5.1 Pressing delete will create confirmation modal

4.4: There will be a tab called "Community Page"

4.4.1: This tab will take the user to the community page

4.4.1.1: This page will display posts by all users marked public

4.4.1.1.2: Post will be able to be flagged by any user, removing them from the community page (no other administration needed)

4.4.1.1.2.1: Post removed from the community page will stay on the user's account

4.4.1.2: Posts on this page will not be editable

4.5: There will be a tab called "Resources"

4.5.1: This tab will take the user to the resource page

4.5.1.1: There will be a label Resources

4.5.1.2 There will be a list of hyperlinks to resources related to mental health

4.6: There will be a tab called "Journal History"

4.6.1: This tab will take the user to their Journal History page

4.6.1.1: see Journal History Component

4.7: There will be a tab called "User Settings"

4.7.1: This button will take users to their personal settings page

4.7.2: see User Setting Component

4.8: There will be a button called "Logout"

4.8.1: This button will log the user out of their account

4.8.1.1: Users will be prompted if they wish to log out or not

5. Journal Entry Component

5.1 There will be a label that prompts a user to enter a headline for the daily journal entry

5.2 There will be a text box that will take a user's headline entry

5.3 There will be a toggle switch for a user to choose the format of the entry as either text or audio

5.3.1 The toggle will be set to text by default

5.3.2 While the toggle is set to text a text box will be visible

5.3.3 The toggle can be set to Audio

5.3.4 When the toggle is set to audio an audio module will be visible

5.3.5 There will be a Record, and a Play Button

5.3.5.1 When record is pressed the audio module will use the user's recording device to record an audio entry. Pressing the button again will stop the recording.

5.3.5.2The play button will play the current recording

5.4 There will be a Post button that allows a user to submit the current entry and headline.

5.4.1 When Post is pressed a dialogue box will pop up to confirm or cancel the post

5.4.2 If confirmed the entry will be sent to the database

5.4.3: The journal entry will be stored in the user's section of the database under their specific username with the headline, date, time, and the content

5.4.4 If canceled the dialogue box will close

5.5 There will be a Cancel Button that allows a user to cancel and delete a current entry.

5.5.1 When submit is pressed a dialogue box will ask to confirm or cancel

5.5.2 If confirmed the current headline and entry will be reset

5.5.3 If canceled the dialogue box will close

5.6: There will be a delete button for journal entries to remove entries from a user's account and the community page

5.6.1: When pressed a dialog box will pop up to confirm or cancel the deletion of the entry

5.6.2: If confirmed a request will be sent to the database attempting to remove the entry

5.6.2.1: A confirmation will pop up if the entry is successfully removed

5.6.3: The entry will be removed from the community page as well if it was visible on the community page

5.7 There will be a public or private toggle button for entries

5.7.1 Private will be the default position

5.7.2 When private is selected posted submissions will not be visible on the community page

5.7.3 When public is selected then posted submissions will be visible on the community page

5.7.3.1: Posts selected to the community page will be sent to the "community page" section of the database with the headline, date, time, and content

6. Word Cloud Component

6.1 There will be a Word Cloud component that visually represents the key words taken from all the user's headlines that have been posted.

6.1.1 The word cloud will be of varying sizes and colors in relation to the number of occurrences of key words.

6.1.2: The headlines will be acquired by making a call to the database and retrieving all headlines from the user's journal entries, then compiling them as a string

6.1.3: The word cloud should utilize dynamic visual tools such as on-hover distinctions

6.1.4: The keywords should individually be clickable

6.1.4.1: Clicking a keyword will navigate to pop up window that will display all posts made by that user that contain that keyword

6.1.4.2: A call to the database will be made to find all headlines matching the clicked selection and puts all entries on the screen

6.1.4.3: A list of article headlines and date stamps and content will be listed in order from most recent.

6.1.4.4: Clicking a headline will populate the Article Display Component

7. Journal History Component

7.1 Will display a list of all a current user's posted articles listed from most recent

7.1.1 Will have label with page name "Journal History"

7.1.2 Will have search bar for keywords in headlines

7.1.3 Will be listed by headlines with Date Stamp included

7.1.4 Will be a web page accessible from navigation

7.1.5: Using the search bar will look through all available entries in the database

7.1.5.1: Will show all articles that are found matching the headline

7.1.5.2: Will tell the user that no entries could be found if no headlines match the search

7.2 Clicking a headline will create a modal instance of the Article Display Component populated with journal headline and content

8. Community Page Component

8.1 Will display a list of all public marked posted articles from all users listed from most recent

- 8.1.1 Will have label with page name “Community Page”
- 8.1.2 Will have search bar for keywords in headlines
- 8.1.3 Will be listed by headlines with Date Stamp included
  - 8.1.3.1: A list of all public entries will be stored in the database and be called to populate the community page in order (see 8.1)
- 8.1.4 Will be a web page accessible from navigation

8.2 Clicking a headline will create a modal instance populated with journal headline and content

- 8.2.1: Will make a call to the database and get the entry from the user page and puts it on the screen
- 8.2.2: Will display a message if the entry is unable to be found in the database (see 5.6 - 5.7.3.1)

9. Admin Page Component

9.1 Will display a list of all User Accounts currently active

- 9.1.1 Will have label with page name “Admin Page”
- 9.1.2 Will be listed by username
  - 9.1.2.1 Will include date created
- 9.1.3 Will have a delete button
  - 9.1.3.1 Clicking delete will create pop-up modal asking to confirm action
  - 9.1.3.2 Confirming delete action will delete that user account
    - 9.1.3.2.1: Sends a request to the database to delete the user account from the database, starting with the username and all information associated with the account
    - 9.1.3.1.2: Deletes all associated entries and information from the community page and “community page” database section
- 9.1.4 Will be accessible under user settings ONLY IF the user is an administrator

10. User Settings Component

10.1 Users will be able to do the following on the page:

- 10.1.1 Change avatar
- 10.1.2 Selecting change avatar will create Modal
  - 10.1.2.1 Labeled Change Avatar
  - 10.1.2.2 Shows current avatar
  - 10.1.2.3Clicking avatar brings up modal to choose image file
    - 10.1.2.3.1 Confirm Button
      - 10.1.2.3.1.1: Uploads the user’s new avatar, and sets the new avatar in the database
    - 10.1.2.3.2 Cancel Button
      - 10.1.2.3.2.1 Closes the change avatar window with no changes

- 10.1.3 Delete account
  - 10.1.3.1 Users will be given a prompt to confirm if they wish to delete their account before doing so

10.1.3.2 If the user clicks confirm, their account will be deleted, along with their posts and username from the database

10.1.3.3 If cancel is selected, the prompt will close

11. Article Display Component

11.1 Will be a Modal that contains:

11.1.1 Article Headline

11.1.2 Article Content

11.1.3 Article Delete Button (If owned by user)

11.1.3.1 Selecting delete will create a confirmation modal

11.1.3.2 Confirmation will delete article from database

12. Mobile Site Component

12.1 Site will be friendly for mobile users

12.1.1 Website will configure containers and information to fit a mobile device

12.1.2 All pages on mobile components will be easily navigated

12.1.3 Journal entries will allow for images to be uploaded from mobile devices

APPENDIX

Figure 1: Developers/Architects

Name	Role
Joel Scott	Developer, Architect
Benjamin Seifried	Developer, Architect

Figure 2: Tech-Stack

Name	Purpose
Node.js	Framework/library to implement server
React	Framework/library to assist with front-end development
Firebase	Database to store information
Replit	Server Host
GitHub	Version Control

Figure 3: Tentative Schedule

Week 1: August 28th – September 6th	**All Members:** Project proposal, assessment of strengths and weaknesses, determination of technology and software, information gathering
Week 2: September 7th – September 13th	**All Members:** Project requirements, refinement of project ideas, setup of technology for project development
Week 3: September 14th – September 20th	**Benjamin:** Focus on front end development, Familiarizing with ReactJS, TypeScript, and JavaScript **Joel:** Focus on familiarizing with back-end technology NodeJS and Firebase
Week 4: September 21st – September 27th	**Benjamin:** Begins to setup overall base of website, sets up front-end environment, and continues to work with ReactJS, TypeScript, and JavaScript **Joel:**

	Begins to setup the database component, setting up back-end environment, and continues working with NodeJS and Firebase
Week 5: September 28th – October 4th	**Benjamin:** Continue working on front-end environment, uses HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets) to create basic design of website **Joel:** Connects database to front-end environment, assists with front-end development, sets up connection for server component of the website
Week 6: October 5th – October 11th	**Benjamin:** Continues working on front-end environment, works on creating UI and creating user journal entry page **Joel:** Assists with front-end, creates data structures to sort and manage user profiles within the server, tests, and debugs current back-end code to ensure quality
Week 7: October 12th – October 18th	**Benjamin:** Begins with word cloud implementation, develops subject categorization **Joel:** Works on back-end development, works on user-authentication through Firebase
Week 8: October 19th – October 25th	**Benjamin:** Continues front-end development, works with team to integrate front-end and back-end together **Joel:** Continues back-end development, works with team to integrate front-end and back-end together
Week 9: October 26th – November 1st	**All Members:** Continue integration between front-end and back-end, begin working on project poster
Week 10: November 2nd – November 8th	**All Members:** Continue working on project poster **Benjamin:** Works on developing navigation bar, creating menu items to navigate around site **Joel:** Continues working on server component, tests and verifies that site has valid connection to server

Week 11: November 9th – November 15th	**All Members:** Continue working on project poster **Benjamin:** Integrates front end and back end, works on UI and ensuring that navigation bar is functioning properly **Joel:** Works with database, ensures that information is being stored and brought to the website
Week 12: November 16th – November 22nd	**All Members:** Finish project poster **Benjamin:** Continue testing features of the website ensure that all features function as intended **Joel:** Test back-end development, ensure that server and database components are working properly
Week 13: November 23rd – November 29th	**All Members:** Continue testing of project, ensure that integration between the components is working
Week 14: November 30th – December 6th	**All Members:** Continue testing and debugging, begin preparing for final interview and project presentation
Week 15: December 6th – December 12th	**All Members:** Project finalization, continue working with group to ensure that they are prepared for the final interview and presentation