Selection of framework/architecture

- Webpage Hosting
  - Google Cloud Run (Website Example)
    - § We will use docker to containerize our front-end tech stack for the desktop version, and host this by deploying it onto cloud run. By doing so, we get the scalability, availability, and security for which our application needs.

- Front-end
  - React & Ionic Framework
    - § Our application requires multi-platform support, and as such, Ionic provides us the tools to cross-compile and utilize one codebase while providing a native-feel across all of the different platforms. As Ionic can integrate with multiple front-end frameworks, we've decided to choose React as our application will be highly component-based, which React tends to perform better with.

- Back-end
  - Google Cloud Run (API Example)
    - § We will use Docker to containerize our API, and then build & deploy it via Google Cloud Run, as it allows us to be hassle-free of scalability concerns and it is fully managed.

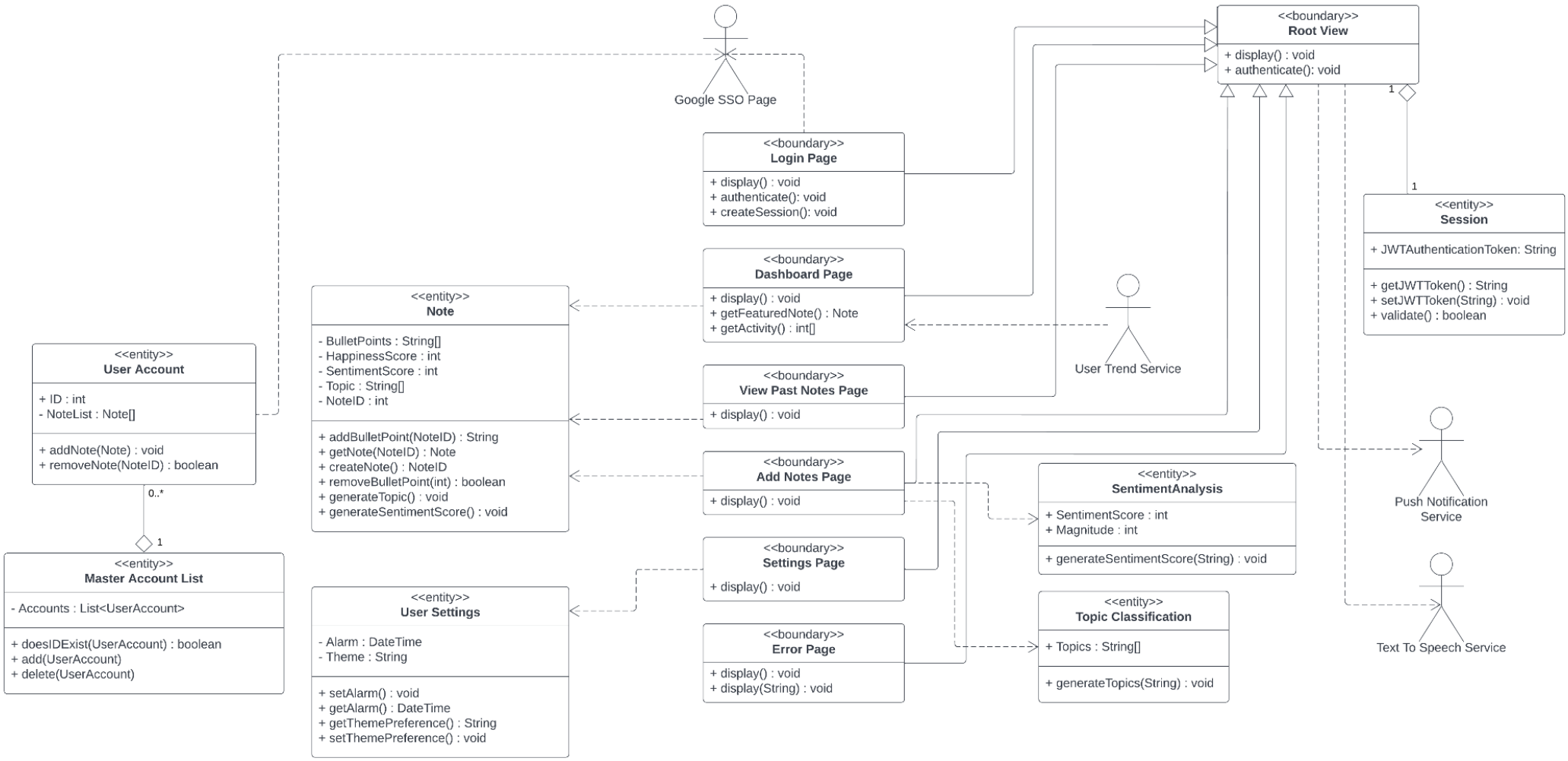
- Machine-Learning Services
  - Sentiment Analysis
    - § Natural Language API (Analyzing Sentiment)
      - Google Cloud allows us to send a string (or from Cloud Storage) to their Natural Language AI and produces a sentiment score and magnitude.
      -
  - Speech-To-Text
    - § Cloud Speech-To-Text
      - To enable Speech-To-Text on computers, and devices without a built-in live transcription, we will use Google Cloud's Speech-To-Text API to send in voice and retrieve text from the end user.
  - Topic Modeling / Text Classification
    - § Natural Language API (Analyzing Entities)
      - Google Cloud also allows us to analyze topics within a string which can be stored and utilized for our Entity Encouragement Service to encourage users to input different entries.
    - § Natural Language API (Analyzing Entity Sentiment)
      - We can use this, in order to draw up relations for our trend service within our application and draw potential connections between an entity and their sentiment value for users.

- Authentication Services
    - Identity Management (Firebase Authentication)
        - § As our application desires to have Google SSO, we've decided to use Firebase Authentication. There exists an integration within the SDK to work with React (our front-end).

- Database Services
    - Images / Large Data Objects (Voice/Video)
        - § Cloud Storage (Google)
            - We plan to use storage buckets as our primary choice of storing user voice/video/image data, as they are incredibly scalable, and the infrastructure is fully managed by google. Furthermore, it has fine-grained permissions to ensure that the data is held securely. It also can easily integrate with our REST API, as itself is also accessed via REST API. Moreover, it has an integration with Firebase Authentication
    - User Data (Notes)
        - § Firestore (Google Firebase)
            - A Serverless No-SQL Database, as our user data will tend to be unstructured data, that is more suited towards a document-based data type.

- Version Control
    - GitHub
        - We are deciding to use GitHub for source control, as it is quite easy to learn and has plenty of documentation and support for any issue that we encounter. Furthermore, there are numerous plugins that allow for seamless integration between our repository and deployment. In addition, it allows us to neatly organize our applications in separate repositories and group it under an organization for free.

Static Class Diagram (including attributes and operations, OO version)



Login

BASIC COURSE:

On the Login Page, the User clicks Sign-in with Google Button, and the system displays Google SSO Page. The User types Password for their Google Account. Then, the User clicks Google SSO Next Button, where Google Identity generates authentication cookies on a valid login. After cookie generation, the system receives the cookies from Google Identity and displays Gratitude Notes Dashboard Page.

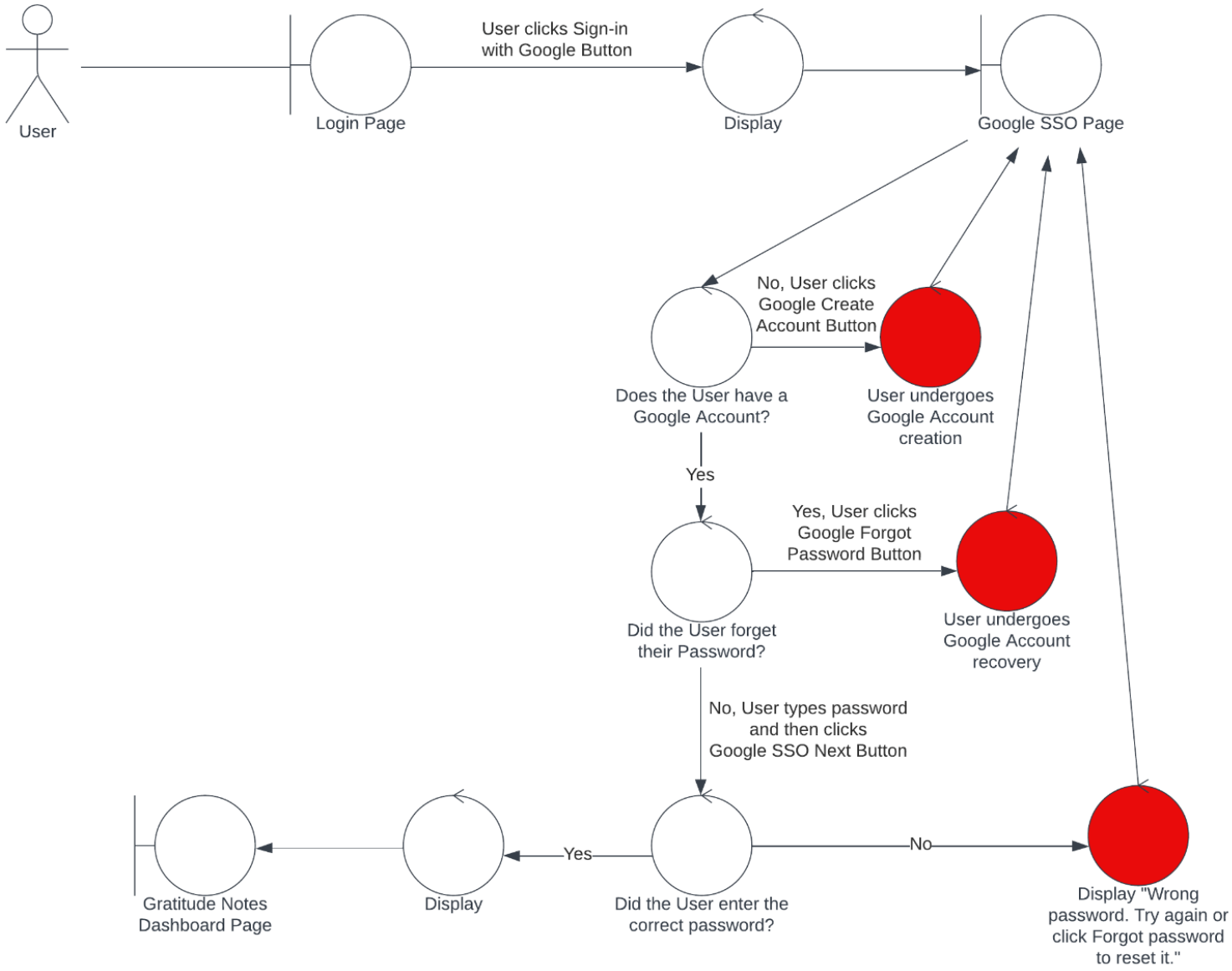
ALTERNATE COURSES:

User does not have a Google Account or they want to create a Google Account: Then, the User clicks Google Create Account Button. Next, the User undergoes Google Account creation. After verification, the system displays Gratitude Notes Dashboard Page.

User forgot their Password: Then, the User clicks Google Forgot Password Button. Next, the User undergoes Google Account recovery. After verification, the system displays Gratitude Notes Dashboard Page.

User enters wrong Password: Then, Google displays Google Wrong Password message, which contains “Wrong password. Try again or click Forgot password to reset it.”

Author: Brady Waughen



Login

BASIC COURSE:

On the Login Page, the User clicks Sign-in with Google Button, and the system displays Google SSO Page. The User types Password for their Google Account. Then, the User clicks Google SSO Next Button, where Google Identity generates authentication cookies on a valid login. After cookie generation, the system receives the cookies from Google Identity and displays Gratitude Notes Dashboard Page.

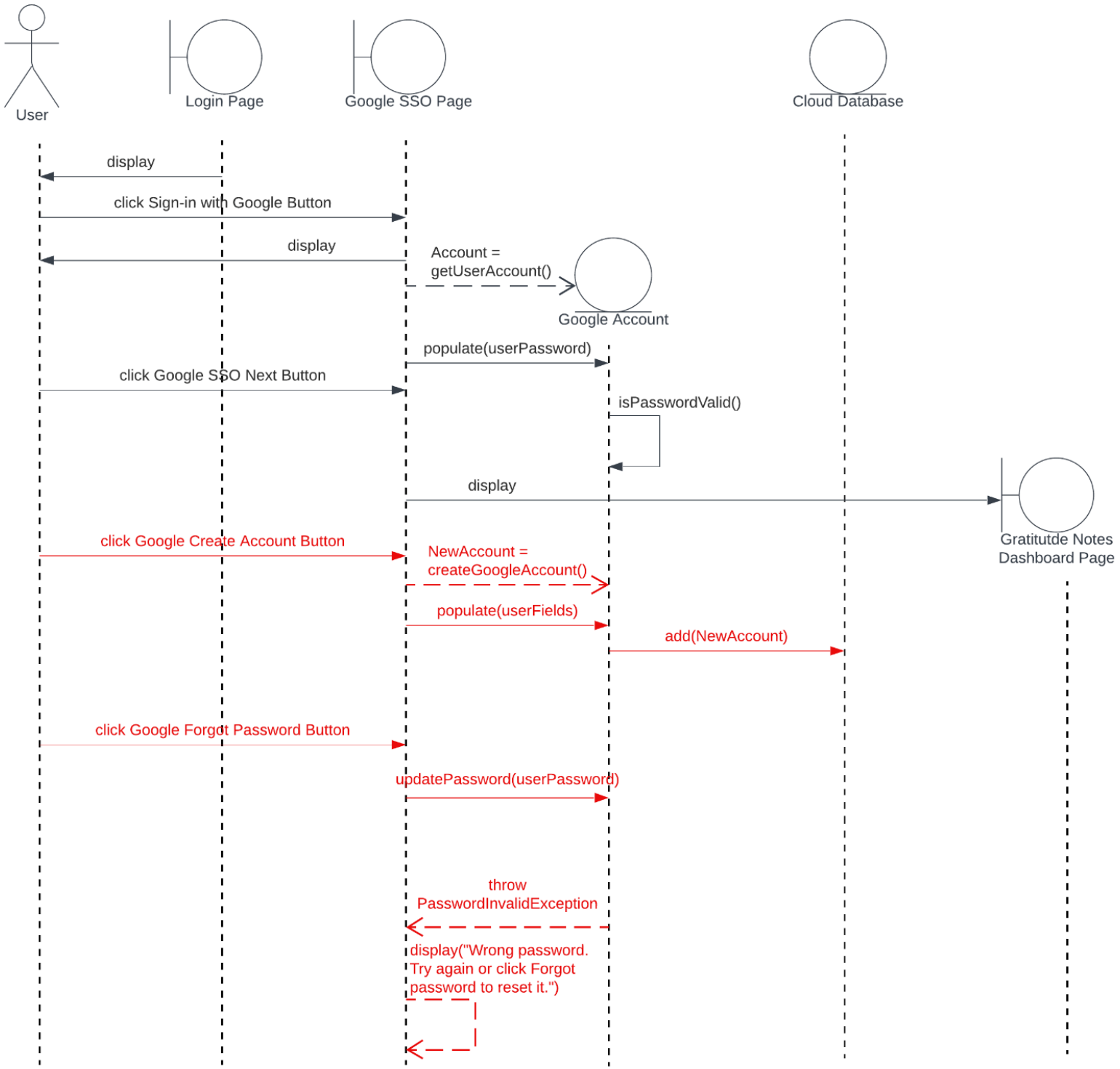
ALTERNATE COURSES:

User does not have a Google Account or they want to create a Google Account: Then, the User clicks Google Create Account Button. Next, the User undergoes Google Account creation. After verification, the system displays Gratitude Notes Dashboard Page.

User forgot their Password: Then, the User clicks Google Forgot Password Button. Next, the User undergoes Google Account recovery. After verification, the system displays Gratitude Notes Dashboard Page.

User enters wrong Password: Then, Google displays Google Wrong Password message, which contains “Wrong password. Try again or click Forgot password to reset it.”

Author: Brady Waughen



Author: Allison Moyer

Add Notes

BASIC COURSE:

On the Gratitude Notes Dashboard Page, the User clicks the Add Button. The system displays Add Notes Page. The User enters their Note Entry and their Gratitude Happiness Score. After, the User clicks the Submit Button; the system checks the Note Entry does not exceed 280 words and the Happiness Score. Then, the system sends the Note Entry to the Entry Encouragement Service. After the check, the system updates the Cloud Database and the system sends the Note Entry to the Text Classification Service and Sentiment Analysis Service. Lastly, the system displays the Gratitude Notes Dashboard Page.

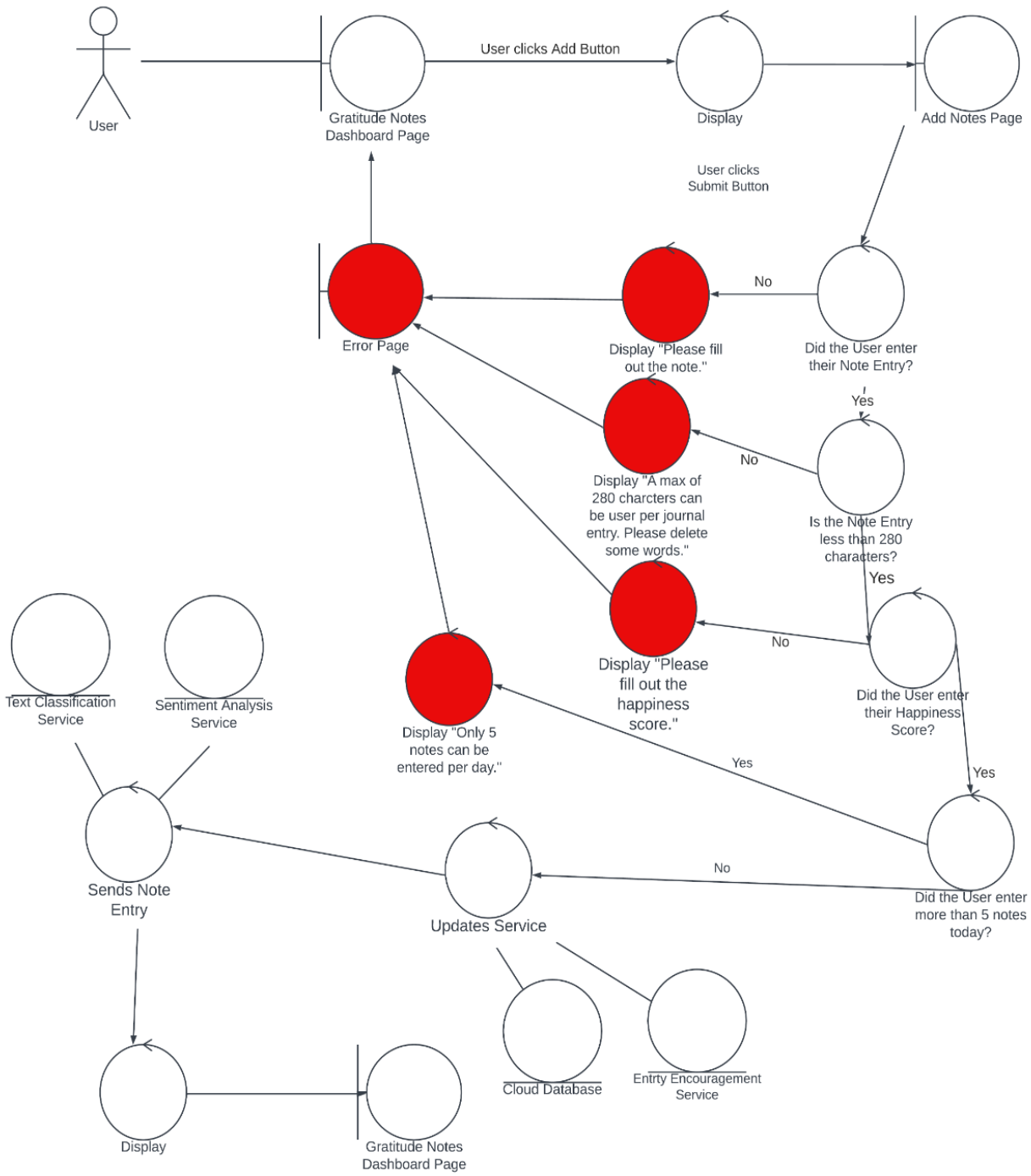
ALTERNATE COURSES:

User hits the submit button without entering a note: The system shows an Error Page, which contains, "Please fill out the note." Then, the system displays Gratitude Notes Dashboard Page.

User hits the submit button without entering a happiness score: The system shows an Error Page, which contains, "Please fill out the happiness score." Then, the system displays Gratitude Notes Dashboard Page.

User fills out more than 5 notes: The system shows an Error Page which contains a message, "Only 5 notes can be entered per day." Then, the system displays Gratitude Notes Dashboard Page.

The Note Entry exceeds more than 280 characters: The system shows an Error Page, which contains a message, "A max of 280 characters can be used per journal entry. Please delete some words." Then, the system displays the Gratitude Notes Dashboard Page.



Author: Allison Moyer

Add Notes

BASIC COURSE:

On the Gratitude Notes Dashboard Page, the User clicks the Add Button. The system displays Add Notes Page. The User enters their Note Entry and their Gratitude Happiness Score. After, the User clicks the Submit Button; the system checks the Note Entry does not exceed 280 words and the Happiness Score. Then, the system sends the Note Entry to the Entry Encouragement Service. After the check, the system updates the Cloud Database and the system sends the Note Entry to the Text Classification Service and Sentiment Analysis Service. Lastly, the system displays the Gratitude Notes Dashboard Page.

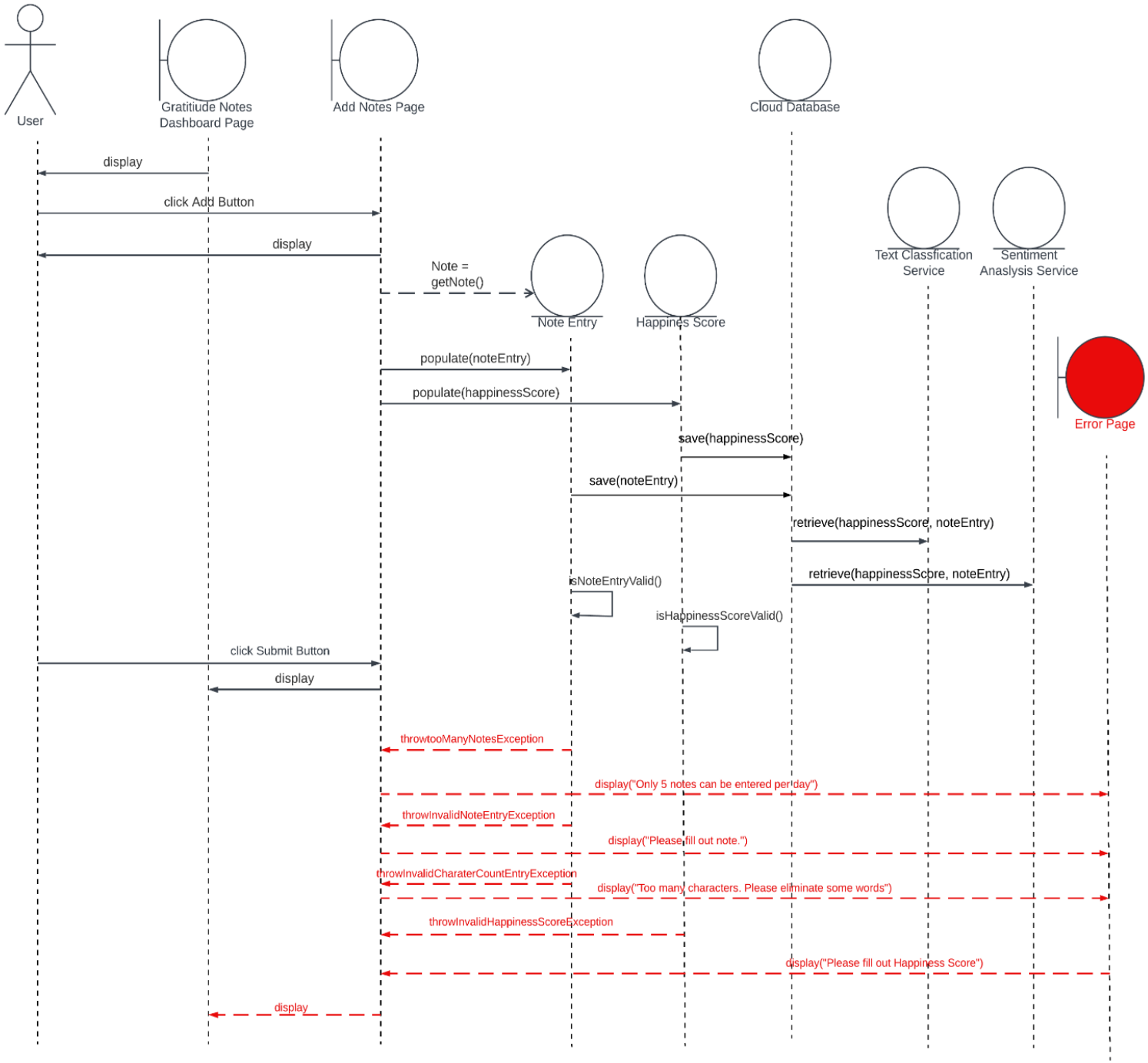
ALTERNATE COURSES:

User hits the submit button without entering a note: The system shows an Error Page, which contains, "Please fill out the note." Then, the system displays Gratitude Notes Dashboard Page.

User hits the submit button without entering a happiness score: The system shows an Error Page, which contains, "Please fill out the happiness score." Then, the system displays Gratitude Notes Dashboard Page.

User fills out more than 5 notes: The system shows an Error Page which contains a message, "Only 5 notes can be entered per day." Then, the system displays Gratitude Notes Dashboard Page.

The Note Entry exceeds more than 280 characters: The system shows an Error Page, which contains a message, "A max of 280 characters can be used per journal entry. Please delete some words." Then, the system displays the Gratitude Notes Dashboard Page.



View Past Notes Robustness Diagram

Author: Myan Nguyen

View Past Note Use Case Text

BASIC COURSE:

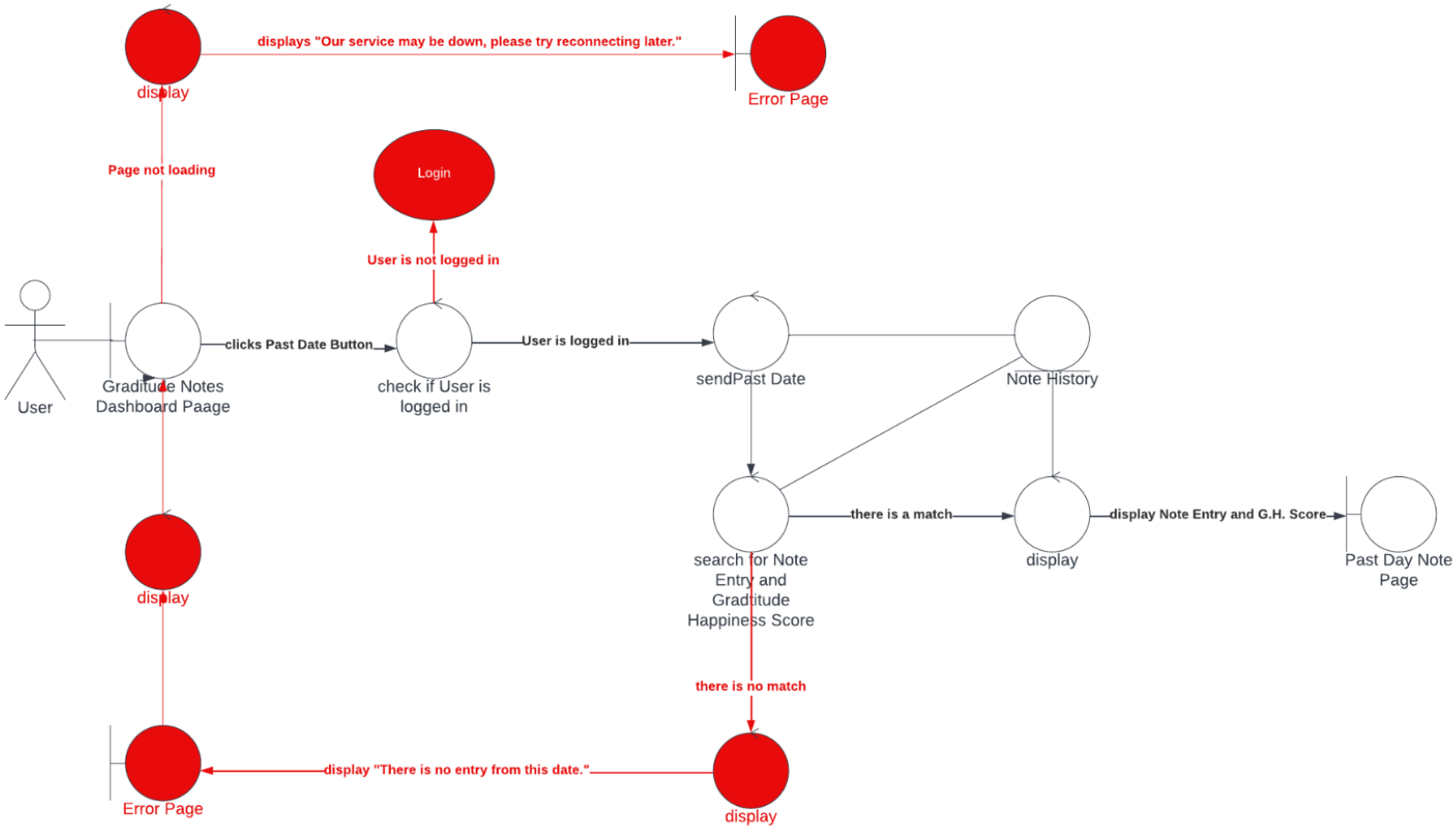
On the Gratitude Notes Dashboard Page, the User clicks Past Date Button from Calendar Widget. The system receives Past Date and the system searches for Note Entry and Gratitude Happiness Score from Note History. The system displays Note Entry and Gratitude Happiness Score on the Past Day Note Page.

ALTERNATE COURSES:

Page not loading: The system shows Error Page, which contains a message, "Our service may be down, please try reconnecting later."

A past date has no matching Gratitude Note/Happiness Score: The system shows Error Page, which contains a message, "There is no entry from this date." Then, the system displays Gratitude Notes Dashboard Page.

The User is not logged in: The session invokes the Login Use Case.



View Past Notes Sequence Diagram

Author: Myan Nguyen

View Past Note Use Case Text

BASIC COURSE:

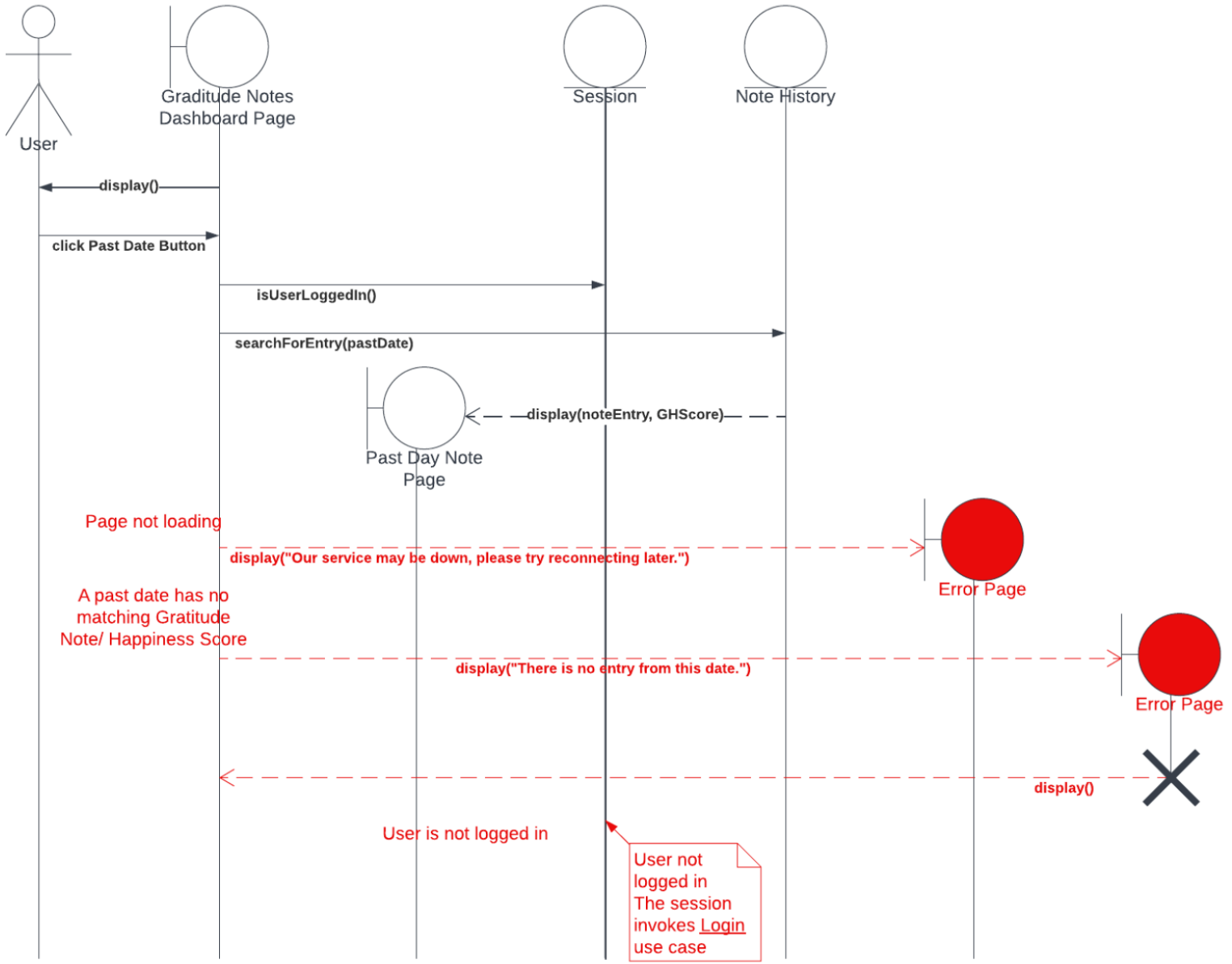
On the Gratitude Notes Dashboard Page, the User clicks Past Date Button from Calendar Widget. The system receives Past Date and the system searches for Note Entry and Gratitude Happiness Score from Note History. The system displays Note Entry and Gratitude Happiness Score on the Past Day Note Page.

ALTERNATE COURSES:

Page not loading: The system shows Error Page, which contains a message, "Our service may be down, please try reconnecting later."

A past date has no matching Gratitude Note/Happiness Score: The system shows Error Page, which contains a message, "There is no entry from this date." Then, the system displays Gratitude Notes Dashboard Page.

The User is not logged in: The session invokes the Login Use Case.



Software Development Plan (Weekly software process schedule)

Month	Day	Development Plan	Test & Demonstration Plan
Jan	15	● Figma prototype On-boarding ● Figma prototype of sign in page ● Figma prototype of settings page ● Figma prototype of adding notes	● Mock-up User Interaction Flow for Sign-In & Dashboard & Adding Notes & New User On-boarding
	22	● Figma prototype of viewing past notes ● Figma prototype of dashboard page ● Final Design Review	● Mock-up User Interaction for Viewing Past Notes & Modifying User Settings
	29	● Create a GCP Account and Establish our Project ● Setup IAM to allow team members to utilize cloud services ● Setup Budget Restrictions & Alerts ● Create Initial Routes and Pages (Just setting up endpoints) ● Create Initial Route & Page Tests	● Ability to view the basic endpoint on a local system ● Team members are able to utilize google cloud services

Feb	5	• Setup Github Actions Initial Route & Page Test Automation • Deploy Initial Front-end Pages on Google Cloud App Engine • Create Initial document-schema (Firestore) [w/ mock values for sentiment analysis and other AI results]	• User can view endpoints on production website (No UI/Functionality yet)
	12	• Create Initial API (Simple GET Endpoints) • Create Initial API Tests • Setup Github Initial API Test Automation • Deploy Initial API on Google Cloud App Engine • Create Initial API Documentation through OpenSwagger	• User can test status of website
	19	• Setup Firestore Authentication within GCP Project • Develop fully-fledged version of Sign-In Page (callback to dashboard page, and ensure redirect if user is not authenticated) • Develop Sign-In Page Tests • Integrate Sign-In Test Automation • Deploy Sign-In Page on Google Cloud App Engine	• Users can interact with sign-in flow and register their account.

	26	• Develop Dashboard-Related Authenticated API endpoints (AI fields will rely on mock values from mock user) • Document new Authenticated Endpoints through OpenSwagger • Develop Dashboard-Related Authenticated API Tests • Document all endpoints within code • Integrate Dashboard-Related Authenticated API to Test Automation • Deploy Dashboard-Related Authenticated API	• Users can view their dashboard information (JSON Format)
March	5	• Develop fully-fledged version of Dashboard Page (Based on mock user field values) • Develop Dashboard Page Tests • Integrate Dashboard Test Automation • Deploy Dashboard Page on Google Cloud App Engine	• Users can view their dashboard for their featured moment (GUI)
	12	• Develop Add Notes-Related Authenticated API endpoints • Document new Authenticated Endpoints through OpenSwagger • Develop Add Notes-Related Authenticated API Tests • Document all endpoints within code • Integrate Add Notes-Related Authenticated API to Test Automation • Deploy Add Notes-Related Authenticated API	• Users can test to add notes via API call on OpenSwagger. (Prototype JSON / HTML)

	19	• Initialize Google Cloud NLP API within GCP Project • Develop fully-fledged version of Add Notes Page • Develop Add Notes Page Tests • Integrate Add Notes Page Test Automation • Deploy Add Notes Page on Google Cloud App Engine	• Users can test/do if they can add notes through the application GUI.
	26	• Develop View Notes-Related Authenticated API endpoints • Document new Authenticated Endpoints through OpenSwagger • Develop View Notes-Related Authenticated API Tests • Document all endpoints within code • Integrate View Notes-Related Authenticated API to Test Automation • Deploy View Notes-Related Authenticated API	• Users can test if they can retrieve their notes via API call on OpenSwagger (Prototype JSON/HTML)
April	2	• Develop fully-fledged version of View Notes Page • Develop View Notes Page Tests • Integrate View Notes Page Test Automation • Deploy View Notes Page on Google Cloud App Engine	• Users can test/do if they can view past notes through the application GUI.

	9	• Develop fully-fledged version of Settings Page • Develop Settings Page Tests • Integrate Settings Page Test Automation • Deploy Settings Page on Google Cloud App Engine	• Users can modify settings and it will be saved locally on their device.
	16	• Create Product Page for Capstone Conference	• Users can view product page and determine if our application is suited for them.
	23	• Create Product Presentation for Capstone Conference	

Software Development Plan (Gantt Chart)

	Jan			Feb				March				April			
	16	23	30	6	13	20	27	6	13	20	27	3	10	17	24
Figma Prototypes															
Google Cloud Environment & Initial Frontend Pages / Routes															
Sign-In Flow / User Registration & On-boarding															
User Dashboard															
Add Notes															
View Notes															
Settings Page															
Product Page															
Product Presentation															