Sprint 1 Planning Document

VENote, Team 14
Killian Le Clainche, Jarett Lee, Aaron Neustedter, Chad Stucky, Ryan Sullivan, Mehul Patel

Sprint Overview

During this sprint our team will get the most basic features of the project running. The selected user stories which will lay a solid foundation for both the frontend and backend, implementing the core features of our application. These core features include notebook creation and viewing, page signing, entry submission ui**,** login and text submission. The core developer features that have to be completed include production and development server environment setup and database setup.

Scrum Master: Aaron Neustedter

Meetings Schedule: Tuesdays & Thursdays (1:30 pm - 2:45 pm)

Risks & Challenges

This sprint will set the foundation of our project, so is is very important that everything is implemented correctly. Failing to implement a strong foundation will hinder our ability to maintain our development speed into the following sprints. We will be challenged to stay focused and detail oriented in order to make sure the foundation is correctly implemented.

Current Sprint Details

User Story #1

As a user, I would like to create and view a new notebook

#	Task Description	Estimated Time	Owner
1	Create client UI for creating a notebook	3hr	Chad
2	Create client UI for viewing a notebook	3hr	Chad
3	Create client UI for viewing a page	3hr	Chad
4	Fill the notebook view with notebook data	2hr	Chad
5	Fill the page view with page data	2hr	Chad
6	Transfer notebook data from server to client	3hr	Aaron
7	Transfer notebook data from database to server	1hr	Jarett
8	Update the database to contain new notebooks	1hr	Jarett

Acceptance Criteria

- From the ‘notebook selection’ view, clicking ‘new notebook’ will bring the user the ‘page selection’ view without any pages.
- From the ‘notebook selection’ view, clicking an existing notebook will bring the user the ‘page selection’ view with a list of pages.
- From the ‘page selection’ view, clicking ‘new page’ will bring the user the ‘page editing’ view. Going back to the ‘page selection’ view should show a new page in the list of pages.
- From the ‘page selection’ view, clicking an existing page will bring the user the ‘page’ view.
- From the ‘page selection’ view, clicking ‘new page’ will send a message to the server to update the database to contain a new notebook, verified by checking the database.

User Story #2
As a user, I would like to view all my notebooks

#	Task Description	Estimated Time	Owner
1	Create client UI for viewing a clickable list of all existing notebooks objects for a user	3hr	Chad
2	Fill the notebook selection view with user data	3hr	Chad
3	Transfer user data from server to client	2hr	Aaron
4	Transfer user data from database to server	2hr	Jarett

Acceptance Criteria

- Given the user is logged in, when they go to ‘notebook selection’ view, they will see a UI displaying all their notebooks.

User Story #3
As a user, I would like to sign pages

#	Task Description	Estimated Time	Owner
1	Create client UI for adding signatures to pages	3hr	Chad
2	Update page data with the signature	2hr	Chad
3	Flag data model upon finalization and prepare an object to be synced to server with signature	4hr	Chad
4	Create UI to finalize signatures	2hr	Chad
5	Transfer page data from client to server	1hr	Aaron
6	Transfer user data from server to database	1hr	Jarett

Acceptance Criteria

- Given I can see a currently selected page from notebook, when I look beside the page display UI, then I will see a UI button which signifies to a user the ability to sign their page and stop further edits.
- Given I can see a currently selected page from notebook, when I click on UI button to sign page, then local model for page should be updated with user signature and unique id/login for user.

- Given I can see a currently selected page from notebook, when I click on UI button to sign page, then local data model for page flagged for synchronization and data model prepared for being updated with server.

User Story #4
As a user, I would like to create a new entry in a notebook

#	Task Description	Estimated Time	Owner
1	Create the edit entry view	2hr	Ryan
2	Create submission button and submission process	2hr	Ryan
3	Handle and transfer entry data from server to database	3hr	Jarett
4	Handle and transfer entry data from client to server	3hr	Aaron
5	Validate entry data passed by the connection	3hr	Mehul

Acceptance Criteria

- From the notebook view, clicking ‘create entry’ will show the ‘edit entry’ view.
- From the ‘edit entry’ view, clicking the submission button should show the ‘notebook’ view and the database should update.

User Story #5
As a user, I would like to add text to an entry

#	Task Description	Estimated Time	Owner
1	Create a text input field GUI component	2hr	Ryan
2	Parse into working document in proper format	2hr	Ryan
3	Update local entry data	2hr	Ryan
4	Validate text entries	2hr	Mehul

Acceptance Criteria

- From the ‘edit entry’ view, clicking ‘add text’ will add another text input component and update the local entry data.

User Story #6
As a user, I would like to add an image to a data entry

#	Task Description	Estimated Time	Owner
1	Create an image input field GUI component	2hr	Ryan
2	Update local entry data	2hr	Ryan
3	Create API call to send image data to server	3hr	Jarett
4	Create system to render images with anchors in notebook viewer	3hr	Ryan
5	Validate image data entries	3hr	Mehul

Acceptance Criteria

- Given I am in the ‘edit entry’ view, clicking ‘add image’ will add another image input component for text
- Given I have added an image, the system will update the local entry data.
- Given I have clicked the ‘submit entry’ button, image data will be sent to the server and saved in the database.
- Given I am in the notebook viewer, images will be properly rendered and displayed next to the appropriate text

User Story #7
As a user, I would like to login to my account

#	Task Description	Estimated Time	Owner
1	Form submission for logging in	2h	Killian
2	Recover account form on same page located on same form	2h	Killian
3	Dispatch email to company after user registers that will confirm account	4h	Mehul
4	Blur to next view while verifying account	4h	Killian
5	Set up Firebase Authentication	2h	Jarett
6	Integrate Firebase Auth with our Log in UI form	2h	Jarett

Acceptance Criteria

- Given I can see a login form on entering the site. When I look at the fields I will have a username, password, forgot password, create account, sign in fields.
- Given I can see the create account form after clicking register account. When I look at the fields I will have username, password, company, email, company code, create account.
- Given I can see a login form on entering the site. When I click the sign in button I will be shown a loading screen transitioning into the notebook view system.
- Given I can see the create account form after clicking register account. When I click the create account button the server will dispatch a confirmation email to the company I had typed in after verification the company code and company name matched.

User Story #8

As a user, I would like to publish an existing page to a notebook

#	Task Description	Estimated Time	Owner
1	GUI pop-up with modular form	2h	Killian
2	Update DB values to associate page with notebook	3h	Jarett
3	Confirmation GUI pop-up	1h	Killian
4	Server parses page and crosses out whitespace + finalizes content	5h	Aaron

Acceptance Criteria

- Given I can see the GUI for publishing. When I enter in the modular form data fields and click submit, I should expect the data is verified and pushed into the database.
- Given I can see the GUI for publishing. When I click submit, I should expect the server will process the page into a final state and return the page to the client.

User Story #9
As a manager, I would like to co-sign pages

#	Task Description	Estimated Time	Owner
2	Server verification of proper user dispatching push notification	1h	Aaron
3	Push notification to manager	2h	Aaron
4	Push notification to manager client with link to proper page and GUI pop-up ready	4h	Killian
5	Tests end-to-end for push notification	1h	Killian
6	Co-sign authoring mode of pages	2h	Killian

Acceptance Criteria

- Given I can see the Push notification from the server. When I click the link provided in it, I should expect to be brought to a page with the proper cosign mode enabled and review mode in progress for me to cosign.
- Given I have clicked submit on publish page. When I see that the server has verified the publish action, I should expect my manager to receive a push notification of the new update to the notebook.

User Story #10
As a developer, I would like to have a working database

#	Task Description	Estimated Time	Owner
1	Research and setup firebase	1hr	Jarett
2	Establish connection between firebase and the server	2hr	Jarett
3	Ensure other developers can view/edit the firebase database	1hr	Jarett
4	Fill firebase with placeholder data for testing	1hr	Jarett
5	Setup secure firebase rules	6hr	Mehul

Acceptance Criteria

- When a developer accesses firebase, he can view and edit all appropriate settings
- When a developer accesses the test database, he can manipulate values and change them as needed
- An unauthorized user can't access firebase

User Story #11

As a developer, I would like separate development and production spaces

#	Task Description	Estimated Time	Owner
1	Create a development git repository with appropriate branches	1hr	Aaron
2	Serve the development page from the development	2hr	Aaron
3	Create release system from the development to the production	5hr	Aaron
4	Ensure both branches are independently present, accessible, and working on the server	1hr	Aaron

Acceptance Criteria

- When the development is released, the production version performs the same as the development without altering the database
- Both development and production branches are clearly marked

User Story #12

As a user, I would like to open a secure connection with the server

#	Task Description	Estimated Time	Owner
1	Implement websocket on the client	4hr	Killian
2	Implement websocket on server	4hr	Jarett
3	Setup websocket secure (wss)	3hr	Mehul

Acceptance Criteria

- The client is able to messages to the server and vice versa
- When messages are sent to the server using the websocket secure (wss) protocol from a test websocket client, the same message is received on the server

User Story #13

As a developer, I would like to serve a landing page

#	Task Description	Estimated Time	Owner
1	Implement port forwarding of port 80 and 443 to the server	1hr	Aaron
2	Acquire and setup a domain name and point it to the server	1hr	Aaron
3	Implement a web serving application to serve the landing page	2hr	Aaron
4	Implement HTTPS protocol for all web serving	3hr	Ryan

Acceptance Criteria

- Given the web address to the server, when I type it into my browser, a place holding landing page appears
- When I visit the web address, the connection will be done over port HTTPS
- When I visit the web address, loading takes less than 2 seconds

User Story #14

As a manager, I would like to prevent notebook tampering

#	Task Description	Estimated Time	Owner
1	Determine when to prevent further manipulation based on the User’s work day - backend	3hr	Mehul
2	Create UI to allow engineers to cross out previous entries	2hr	Ryan
3	Create API call to allow entries to be retroactively deleted on the server	2hr	Ryan
4	Create system to render crossed out pages	3hr	Ryan

Acceptance Criteria

- Given I have a page loaded in the notebook viewer, I should be able to select a data entry to cross out
- Given I have selected a data entry to cross out, the UI should warn me that the process is irreversible
- Given that I have clicked the confirm button to cross out a page, the server should mark that page as crossed out for future rendering

User Story #15

As a user, I would like to know the timestamp of entries

#	Task Description	Estimated Time	Owner
1	Create client UI to display the entry creation time in the entry editing view	1hr	Ryan
2	Create client UI to display the entry creation time in the entry view	2hr	Ryan
3	Modify entry data structure to keep track of time stamps and validation	2hr	Jarett
4	Store the entry creation time from the server	2hr	Jarett
5	Modify page validation information	2hr	Mehul

Acceptance Criteria

- On the entry creation view, the user should see an accurate entry creation time
- On the page view, the user should see an accurate entry creation time

User Story #16

As a user, I would like to query notebooks by timestamp

#	Task Description	Estimated Time	Owner
1	Create client UI for querying notebook metadata	4hr	Killian
2	Create user data parsing method	4hr	Killian
3	Create metadata search for notebooks	2hr	Killian

Acceptance Criteria

- Given I am using the search bar, typing in a date should present a sorted list of all entries matching that date.
- Given I click on a result from the search bar, I should be taken to a notebook view that properly displays the entry I selected.

User Story #17

As a user, I would like to export the notebooks as PDFs

#	Task Description	Estimated Time	Owner
1	Collection of rendered pages on backend	4hr	Mehul
2	Compilation into one PDF and dispatches via encrypted line	5hr	Mehul

Acceptance Criteria

- Given the notebook is finished, when I click compile notebook to PDF I should expect a PDF transferred via a secure line to me.
- Given the compile notebook button is clicked, I should expect the server to process all pages into one large PDF with all collected data contained inside.

User Story #18

As a user, I would like to tag notebooks

#	Task Description	Estimated Time	Owner
1	Widget into submission of page with unique tag additions	2hr	Killian
2	Widget into edit notebook with unique tag additions	1hr	Killian
3	Updating page into database with proper tags	2hr	Mehul

Acceptance Criteria

- Given the publish GUI is open, when I add unique tags into the proper field, I should expect the page to now have these tags in the database.
- Given the edit notebook GUI is open, when I add unique tags in the proper field, I should expect the notebook to now have these tags in the database.

Remaining Backlog

Functional

As a user, I would like to

~~1. Create new data entry~~
~~2. See all notebooks/pages I have access to~~
~~3. Login to my account~~
~~4. Publish page~~
~~5. Add text to the notebook~~
~~6. Comment on pages~~
7. As a user I would like an introductory tutorial
8. Make corrections in the proper format
~~9. Add images to the notebook~~
~~10. Sign pages~~
11. Caption pictures in the notebook
12. Use a paintbrush tool to add custom drawings to the notebook
13. Share a notebook through email.
14. Share a notebook through a link.
15. Share a notebook to another person without an account.
~~16. Know the number of pages~~
~~17. Know the timestamp of pages~~
18. Query all notebooks for a specific notebook
19. Query all notebooks for a specific timestamp
20. Query all notebooks for specific text

21. Query a notebook for a specific page
22. ~~Query a notebook for a specific timestamp~~
23. Query a notebook for specific text
24. Query a page for specific text
25. Create notebook groups
26. Share a notebook
27. Give feedback to developers
28. Upload general files to the notebooks
29. Upload STL to the notebooks if time allows
30. Upload FBX to the notebooks if time allows
31. Export notebooks as PDFs containing 3D-viewers
32. Tag notebook pages
33. Export notebooks as PDF
34. Back up a notebook to a secondary storage location
35. Know why I can’t edit previous pages
36. ~~Access the notebook from the internet~~
37. Access the notebook offline if time allows
38. Choose what formatting and rules are applied to my page
39. Location to be logged and have the option to keep track of where I edit my pages from.

As a manager, I would like to

40. ~~Prevent notebook tampering~~
41. View a group of user’s notebooks
42. Co-sign notebook pages
43. Assign user permissions to other users
44. Submit a notebook for legal approval
45. Assign specific users access to certain notebooks
46. Handle a large number of users

Non-Functional

- Intuitive UI - it should be easier for customers to use our tool that it is for them to use existing tools
- Fast site loading times (under 1 second)
- Handle large usage load
- Handle large number of users
- ~~Secure communication line using https~~
- ~~Secure communication line using a direct connection~~
- ~~Developers maintain server backend using Node.js~~
- ~~Developers maintain a secure database using Firebase~~
- ~~Developers maintain a querying mechanic to interact with Firebase database~~
- Developers maintain config file for easy minor tweaks
- Developers alter the server through a centralized location