****

**Team 31 – Design Document**

Pooja Tewari, Shivangi Chand, Siddharth Dhar, Sripath Mishra, Zachary Moore

**Sprint​ ​Overview**

Our goal for this Sprint is to complete the initial set and and successfully create user accounts. In addition, we also aim on completing the initial process of creating and storing notebooks. We will setup the CI/CD pipeline. First we will setup two Github repositories (one for backend and one for frontend), After which we will setup the IDE’s and install all the requirements (React, Node JS, PostgresQL, etc). We will integrate Docker and CircleCI. Then we will set up the database tables in PostgresQL. We will complete the initial setup by setting up heroku by deploying two instances of code for testing and production. After these steps we will start working on the following user stories by creating Carnet user accounts and provide UI for managing these accounts (i.e. forgot password, information update etc.).

**SCRUM​ ​Master**:​ ​​Sripath Mishra

**Meeting​ ​Plan**:​ ​​Monday 4:30-6:00 PM, Thursday 4:30-6:00 PM

**Risks​ ​and​ ​Challenges**

The biggest challenge for our team is maintaining proper communication and a healthy team atmosphere as it will help to increase productivity. Another challenge would be to have a standard set of protocols for writing code such as linting, git, and other formal rules. Once we overcome these two challenges, we should be able to get through the minor ones such as keeping all the developers on track. The scrum master needs to ensure that all the rules are followed and each developer is assessed equally. There are quite a few risks involved in this sprint. First one would be developers not adhering to the standard rules. Another one would be making sure the repositories, the CI/CD pipeline and the heroku deployment are secure. We must make sure that there are no security vendor abilities while doing the initial set up. In addition, we must make sure that the users’ information is stored securely in order to prevent malicious activities. With our detailed product backlog and design document we will be able to overcome most of these hurdles. If any unforeseen challenge or risk arrives we will have to look at the future requirements before tackling those challenges or risks.

**Current​ ​Sprint​ ​Details**

User​ ​Story​ ​1:

As a user, I would like to be able to register for a Carnet account.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create​ ​User​ ​model​ ​in​ Backend | 2 | Siddharth |
| 2 | Create​ ​User​ ​tables​ ​in​ ​Database | 2 | Sripath |
| 3 | Create​ ​a​ ​signup​ ​page​ ​for​ ​to​ ​insert​ ​a new​ ​user​ ​into​ ​the​ ​system | 3 | Shivangi |
| 4 | Tie​ ​the​ ​three​ ​above​ ​components together​ ​to​ ​complete​ ​user​ ​signup process​ ​(frontend​ ​to​ ​backend​ ​to​ ​DB) | 3 | Shivangi and Sripath |
| 5 | Unit​ ​test-​ ​correct​ ​and​ ​incorrect​ ​input | 3 | Pooja |

Acceptance​ ​Criteria:

* ● Given​ ​that​ ​the​ ​user​ ​creation​ ​process​ ​is​ ​functional,​ ​a​ ​user​ ​model​ ​can​ ​be
* instantiated​ ​with​ ​the​ ​desired​ ​information​ ​in​ ​the​ ​backend.
* ● Given​ ​the​ ​user​ ​creation​ ​process​ ​is​ ​functional,​ ​a​ ​user​ ​instance​ ​can​ ​successfully​ ​be
* inserted​ ​into​ ​the​ ​User​ ​table​ ​in​ ​the​ ​database.
* ● Given​ ​the​ ​user​ ​creation​ ​process​ ​is​ ​functional,​ ​a​ ​user​ ​can​ ​view​ ​a​ ​page​ ​to​ ​securely
* enter​ ​their​ ​information​ ​and​ ​save​ ​it​ ​in​ ​the​ Carnet ​system​ ​base.
* ● Given​ ​the​ ​user​ ​creation​ ​process​ ​is​ ​functional,​ ​a​ ​user’s​ ​information​ ​can​ ​be​ ​sent
* from​ ​the​ ​client,​ ​to​ ​the​ ​server,​ ​to​ ​the​ ​backend​ ​without​ ​data​ ​disruption​ ​or
* exploitation.
* ● Given​ ​the​ ​user​ ​creation​ ​process​ ​is​ ​functional,​ ​as​ ​a​ ​developer​ ​we​ ​should​ ​be​ ​able
* to​ ​test​ ​the​ ​input​ ​and​ ​different​ ​types​ ​of​ ​logins,​ ​and​ ​all​ ​unit​ ​tests​ ​should​ ​pass​ ​when expected.

User​ ​Story​ ​2:

As a user, I would like to be able to login and manage my Carnet account.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create​ ​a login page | 3 | Shivangi |
| 2 | Integrate a [BLANK] authentication process | 2 | Sripath |
| 3 | Encrypt​ ​the​ ​password​ ​(highly​ ​sensitive information)​ ​before​ ​any​ ​transfers​ ​occur | 3 | Sripath |
| 4 | Create​ ​and​ ​manage​ ​a​ ​user’s​ ​session | 3 | Siddharth |
| 5 | Unit​ ​Test​ ​-​ ​correct​ ​and​ ​incorrect​ ​input, unknown​ ​email,​ ​incorrect​ ​password attempt​ ​limits | 3 | Pooja |

Acceptance​ ​Criteria:

* ● Given​ ​that​ ​user​ ​login​ ​and​ ​authentication​ ​is​ ​functional,​ ​a​ ​user​ ​can​ ​easily​ ​access
* the​ ​login​ ​page​ ​for​ Carnet,​ ​and​ ​enter​ ​their​ ​login​ ​credentials.
* ● Given​ ​that​ ​there​ ​is​ ​integration​ ​with​ ​the​ ​is​ ​implemented,​ ​a​ ​user’s​ ​login​ ​will​ ​be
* authenticated​ ​based​ ​on​ ​certain​ ​criteria​ ​like​ ​username/password.
* ● Given​ ​that​ ​user​ ​login​ ​and​ ​authentication​ ​is​ ​functional,​ ​a​ ​user’s​ ​password​ ​will​ ​be encrypted​ ​before​ ​a​ ​login​ ​is​ ​considered​ ​a​ ​success​ ​or​ ​failure.
* ● Given​ ​that​ ​creating​ ​and​ ​managing​ ​a​ ​user’s​ ​session​ ​is​ ​implemented,​ ​a​ ​user​ ​will​ ​be able​ ​to​ ​login​ ​to​ ​their​ ​dashboard.
* ● Given​ ​that​ ​a​ ​unit​ ​test​ ​has​ ​been​ ​created​ ​to​ ​test​ ​for​ ​incorrect​ ​input,​ ​then​ ​the​ ​user
* should​ ​be​ ​able​ ​to​ ​receive​ ​feedback

User​ ​Story​ ​3:

As a user, I would like my password to be reset if I forget it.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Set up a password reset page | 3 | Shivangi |
| 2 | Create a password reset function in the backend | 3 | Siddharth |
| 3 | Allow the user to change the password on the frontend | 3 | Siddharth |
| 4 | Update the database on the backend | 3 | Pooja |
| 5 | Unit​ ​Test​ ​-​ ​correct​ ​and​ ​incorrect​ ​input | 3 | Pooja |

Acceptance​ ​Criteria:

* ● Given that the user currently exists in the database, a user should be able to go to the password reset page
* ● Given that the user currently exists in the database, the password reset function will reset the password
* ● Given that the password reset is functional, the user should be able to reset their password
* ● Given that there is integration with the database, the password should be updated in the database

User​ ​Story​ ​4:

As a user, I would like to be able to easily navigate to a personal Notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Selecting the desired notebook to open | 2 | Siddharth |
| 2 | Fetching data from the database | 2 | Siddharth |
| 3 | Projecting notes on the screen | 3 | Siddharth |
| 4 | Testing and Quality Assurance. | 2 | Sripath |

Acceptance​ ​Criteria:

* ● Given the notebook exists in the database, the user should be able to select and open the desired notebook.
* ● Given the user selects a different notebook, fetch its data from the database.
* ● Given the data is fetched from the database, project the notes onto the screen.

User​ ​Story​ ​6:

As a user, I would like to be able to easily access Carnet across all of my web-enabled devices.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Set up Carnet frontend using react | 3 | Shivangi |
| 2 | Testing over multiple web browser in different screen sizes. | 2 | Pooja |

Acceptance​ ​Criteria:

* ● Given the user uses a laptop the screen and tools are optimized to prevent any distortion in the UI irrespective of the browser used.
* ● Given the user uses a tablet the screen and tools are optimized to prevent any distortion in the UI irrespective of the browser used and without requesting for a desktop site.
* ● Given the user uses a phone the screen and tools are optimized to prevent any distortion in the UI irrespective of the browser used and without requesting for a desktop site.

User​ ​Story​ ​7:

As a user, I would like to be able to create new notebooks.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create​ new notebook​ ​in​ ​database | 2 | Sripath |
| 2 | Create​ ​notebook​ ​model in backend | 2 | Sripath |
| 3 | Allow the logged in user to be able to access the new notebook | 3 | Sripath |
| 4 | Create the UI for a new notebook | 4 | Siddharth |
| 5 | Save the changes in the database | 2 | Sripath |
| 6 | Testing (unit and integration) | 2 | Shivangi |

Acceptance​ ​Criteria:

* ● Given the notebook does not exist in the database, create a new notebook.
* ● Given the creation of the network model in the backend, allow user access to the new notebook.
* ● Given the user opens the new notebook, create the UI for it.
* ● Given the user inputs data onto the new notebook, save the changes in the database.

User​ ​Story​ ​8:

As a user, I would like to be able to create new pages in any notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create a new page in database | 2 | Shivangi |
| 2 | Create the new page model in the database | 2 | Pooja |
| 3 | Allow the user to be able to create and access the new page in a notebook | 3 | Shivangi |
| 4 | Create the UI for the new page | 3 | Pooja |
| 5 | Unit​ ​test-​ ​correct​ ​and​ ​incorrect​ ​input, user is authenticated | 3 | Pooja |

Acceptance​ ​Criteria:

* ● Given that the new page creation is functional, a user should be able to add a new page to the database
* ● Given that the new page creation is functional, the new page should be added to the database
* ● Given that the new page creation is functional, the user should be able to access the new page

User Story 24:

As a user, I would like to be able to make my notebooks private.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create UI for making private/ public | 3 | Siddharth |
| 2 | Backend for private/ public | 2 | Sripath |
| 3 | Create Database model for private/public | 3 | Pooja |
| 4 | Unit Test : Input testing | 2 | Pooja |

Acceptance​ ​Criteria:

* ● Given the user has an account, when they are logged in they should be able to create a private notebook which is not viewable by any other user.
* ● Given the user has an account, when they are logged in they should be able to create a public notebook which is viewable by any other user.
* ● Given the developer has access to the database, they should be able to view if the correct changes are made in the database when a notebook is made public or private.

User​ ​Story​ ​29:

As a user, I would like to have a way to connect to the developers for providing feedback

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create contact us page | 3 | Shivangi |
| 2 | Backend of feedback | 2 | Sripath |
| 3 | Create Database model of feedback | 3 | Shivangi |
| 4 | Testing (unit and integration) | 2 | Siddharth |

Acceptance​ ​Criteria:

* ● Given the user has an account, when they are logged in they should be able to access the contact us page.
* ● Given the user has an account, when they are logged in they should be able to provide feedback and identify bugs.
* ● Given the developer has access to the database, they should be able to view feedback and bugs and work on increasing the user experience.

User​ ​Story​ ​30:

As a Software Developer, I would like to have a continuous development and continuous integration pipeline.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create an CIrcleCI account. | 2 | Sripath |
| 2 | Create a hook for the Github repository. | 2 | Sripath |
| 3 | Create, update and add necessary files for CircleCI (like config.yml) | 3 | Siddharth |
| 4 | Create, update and add necessary files for Heroku (like config.yml) | 3 | Siddharth |
| 5 | Test the CI/CD pipeline | 3 | Pooja |

Acceptance​ ​Criteria:

● Given the developer has git installed, he/she should be able to trigger CircleCI’s integration tests and linting tests by pushing his/her commits onto the remote branch.

● Given the CircleCi is setup and the developer creates a PR. Other developers should be able to rely in the CircleCI tests for compile and build time errors.

* ● Given the master branch has changed by an developer, If the CircleCI’s integration tests and linting tests are passed then the master branch should be automatically deployed to staging.

User​ ​Story​ ​31:

As a Software Developer, I would like to have a Docker container for local testing.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Download docker. | 2 | Shivangi |
| 2 | Set up the the Postgres and API containers through docker-compose.yml. | 2 | Sripath |
| 3 | Set up Dockerfile and link the two containers. | 3 | Sripath |
| 4 | Make other changes in API to integrate Docker. | 3 | Shivangi |
| 5 | Test Docker and push the new code. | 3 | Pooja |

Acceptance​ ​Criteria:

* ● Given the developer has docker installed, he/she should be able to build the codebase by typing “docker-compose up”.
* ● Given the developer has docker installed, he/she should be able to run the unit and integration tests over the codebase by typing “docker-compose up”.

● Given the developer has docker installed, he/she should be able to run the API and test it manually by typing “docker-compose up”.

**​Product​ ​Backlog**

~~1. As a user, I would like to be able to register for a Carnet account.~~

~~2. As a user, I would like to be able to login and manage my Carnet account.~~

~~3. As a user, I would like my password to be reset if I forget it.~~

~~4. As a user, I would like to be able to easily navigate to a personal Notebook.~~

5. As a user, I would like to be able to open any page of my notebook

~~6. As a user, I would like to be able to easily access Carnet across all of my web-enabled devices.~~

7. As a user I would like to be able to create new notebooks.

8. As a user, I would like to be able to create new pages in any notebook.

9. As a user, I would like to be able to insert text in the notebook.

10. As a user, I would like to be able to update text in the notebook.

11. As a user, I would like to be able to delete text in the notebook.

12. As a user, I would like to be able to beautify text in the notebook.

13. As a user, I would like to be able to insert pdf as images in the notebook.

14. As a user, I would like to be able to update pdf images in the notebook.

15. As a user, I would like to be able to delete pdf images in the notebook.

16. As a user, I would like to be able to insert text over pdf images.

17. As a user, I would like to be able to update text over pdf images.

18. As a user, I would like to be able to delete text over pdf images.

19. As a user, I would like to be able to beautify text over pdf images.

20. As a user, I would like to be able to save any changes made on the notebook.

21. As a user, I would like to be able to edit notebooks after they have been saved.

22. As a user, I would like to be able to organize the notebooks in folders.

23. As a user, I would like to be able to share my notebook or make my notebooks public.

~~24. As a user, I would like to be able to make my notebooks private.~~

25. As a user, I would like to be able to search other public notebooks by their titles, owners, etc.

26. As a user, I would like to be able to like, dislike and linear comment about a notebook.

27. As a user, I would like to be able to subscribe to another user or any updates on a notebook.

28. As a user, I would like to be able to edit any notebooks which I have been given access to by the owner.

~~29. As a user, I would like to have a way to connect to the developers for providing feedback~~

~~30. As a Software Developer, I would like to have a continuous development and continuous integration pipeline.~~

~~31. As a Software Developer, I would like to have a Docker container for local testing.~~