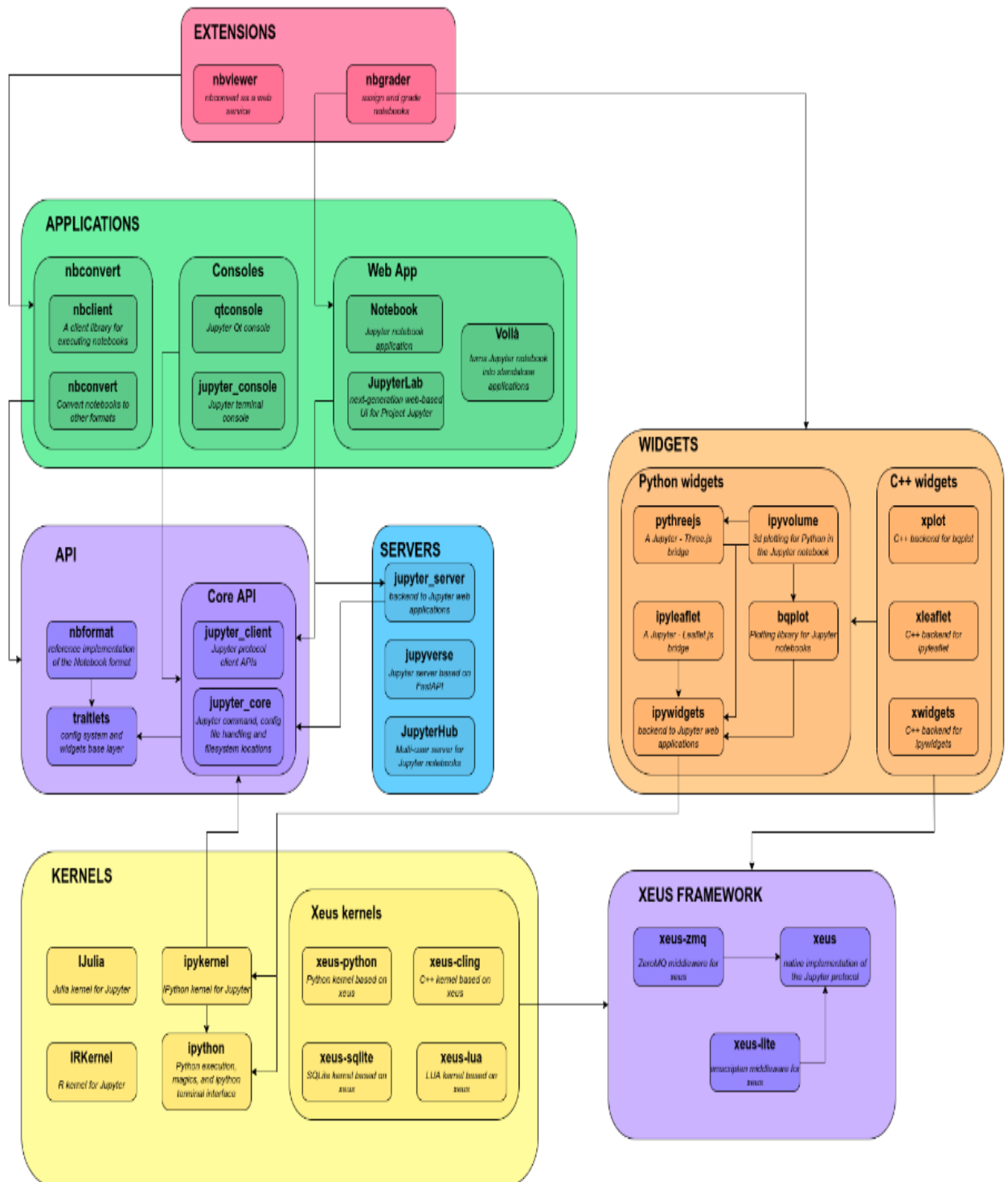# Report analysis For HomeTask4

As per the task assigned below is the required analysis
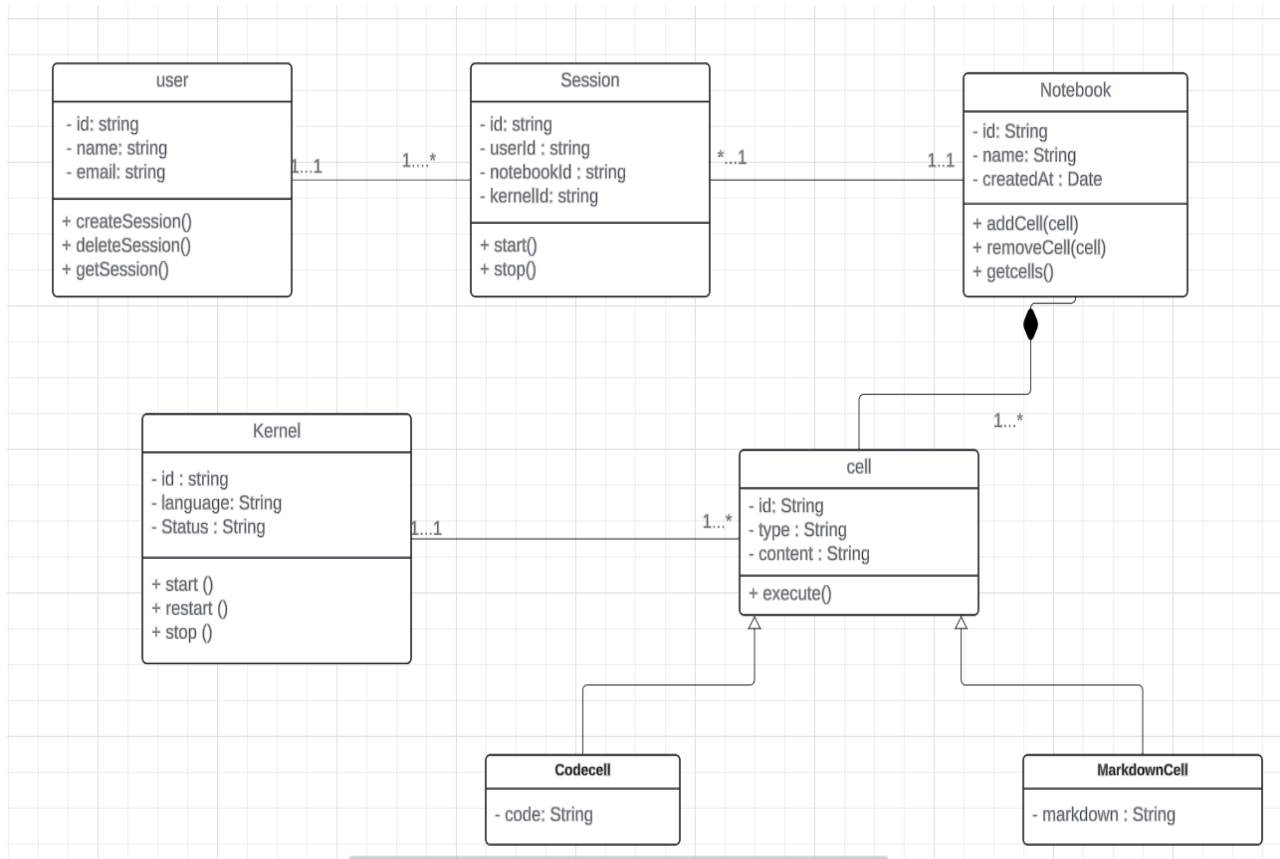
1. **Framework Data Description (internal and external) (Class Diagrams, Component Diagrams)**

Here is the high-level overview of the project relationship.

References:- https://docs.jupyter.org/en/latest/projects/architecture/content-architecture.html

2. Class diagram

https://lucid.app/lucidchart/a3ef2186-2f94-4d3e-94db-136388dde2fc/edit?viewport_loc=-167%2C121%2C2133%2C987%2C0_0&invitationId=inv_86e2f0e0-dd3d-44d4-a9be-e69ede8aaa37

The above diagram is created through the Lucid app here is the link for a direct visit.

Explanation:-

In the above class diagram we have user, session, Notebook, Kernal, cell is the entity in which user have the attributes such as id as unique identifier, name and email id and methods its holds user can create their session, delete their existing session and have the information of get the information of the session with the cardinality one user can create the multiple session.
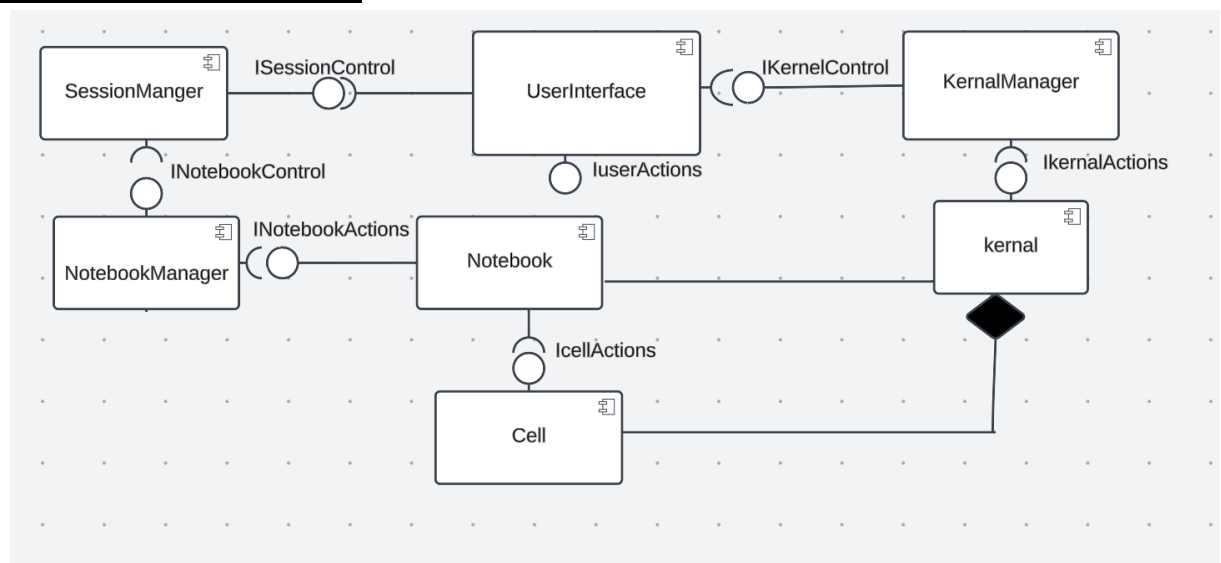
Session our next entity holds the attributes such as user id or it will create in notebook id and kernal id to perform the task. The method it hold such as start the session or stop the session.one session can have only one Notebook id.

Notebook our next entity holds the attributes such as id, name and created at. Contains methods such as add cell, remove cell and get cell as notebook contains cells in it.

Cells are divided into two parts such as code cell and markdown cell these both are inherited by the cell. Cell and notebook have relationship such as without notebook cell does not exist

Kernal as our last entity contains attribute as id, language and status with the methods start, stop and restart.

3. <u>Component diagram</u>

The above diagram is created through the Lucid app here is the link for a direct visit.

<u>Explanation:-</u>

Components and interfaces classification

1. User Interface (UI)
   Provided interfaces:- IuserActions
   Required Interfaces:- IsessionControl, InotebookControl

2. Session Manager
   Provided interfaces:- IsessionControl
   Required Interfaces:- InotebookControl

3. Notebook Manager
   Provided interfaces:- InotebookControl
   Required Interfaces:- InotebookActions

4. Kernal Manager
   Provided interfaces:- IKernalControl
   Required Interfaces:- IKernalActions

5. Notebook

   Provided interfaces:- INotebookActions

   Required Interfaces:- ICellActions
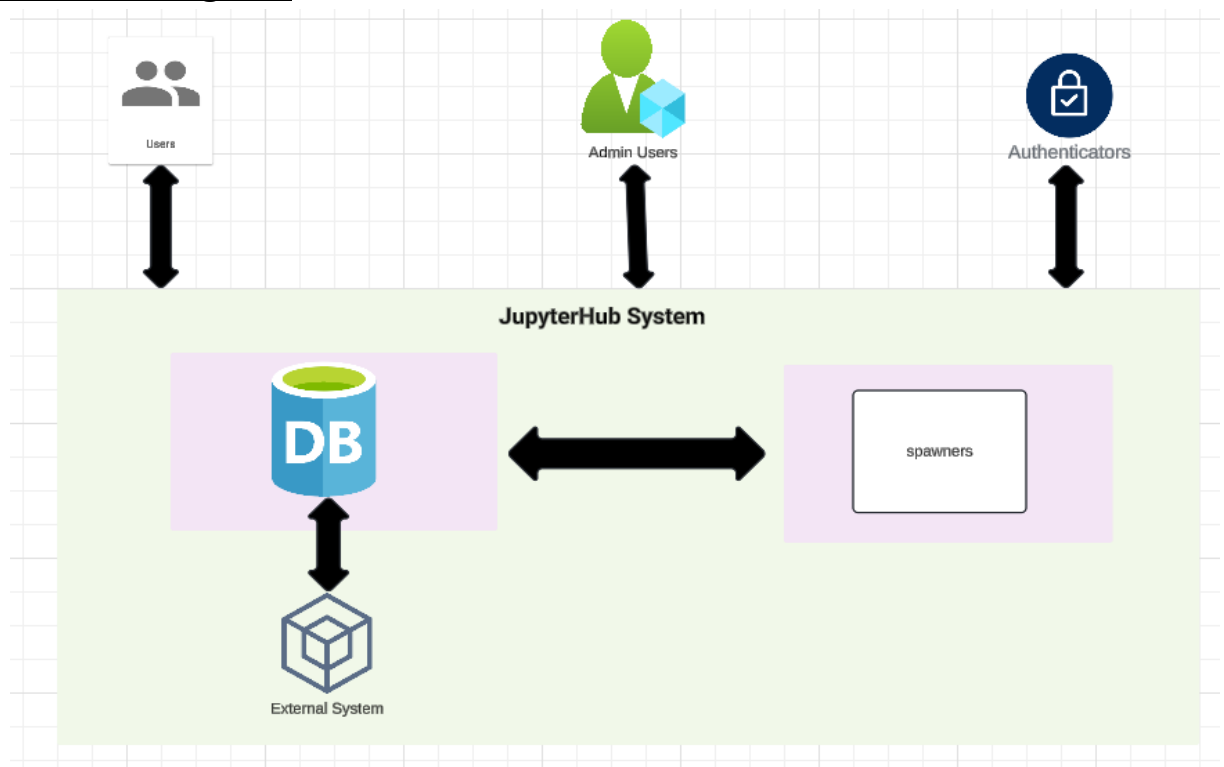
6. Cell

   Provided interfaces:- ICellActions

   Required Interfaces:- IKernalActions

7. Kernal

   Provided interfaces:- IKernalActions

   Required Interfaces:- none

4. Context diagram

The above diagram is created through the Lucidchart app here is the link for a direct visit.

Explanation:-

1. Users: individuals who use the JupyterHub system. They could be students, researchers or any user needing to run jupyter notebooks.
2. Admin user: Adminstrators who manage the jupyterHub System.
3. JupyterHub System: The main System providing the jupyter notebook service.
4. Authenticator: Service used for authenticating user

5. Database: Stores user data, session information, and other necessary data for jupyterHub operations.
6. Spawners: Responsible for spawning jupyter notebook servers for users. Could be local or remote(e.g., Kubernetes, Docker)
7. External Systems: Any other systems or Services JupyterHub interacts with for more functionality or any features.
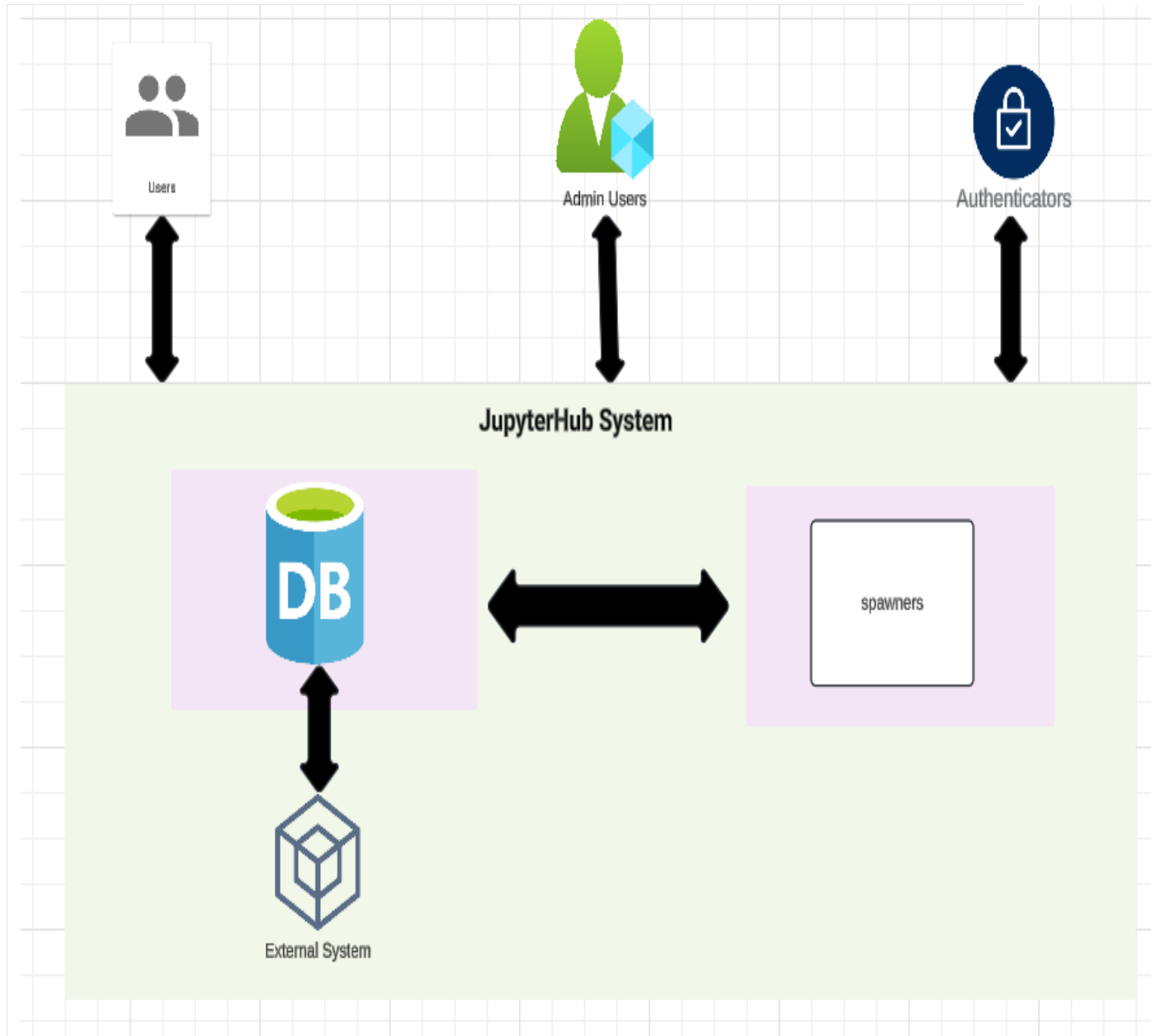
The above diagram is created through the Lucid app here is the link for a direct visit.

5. Deployment diagram
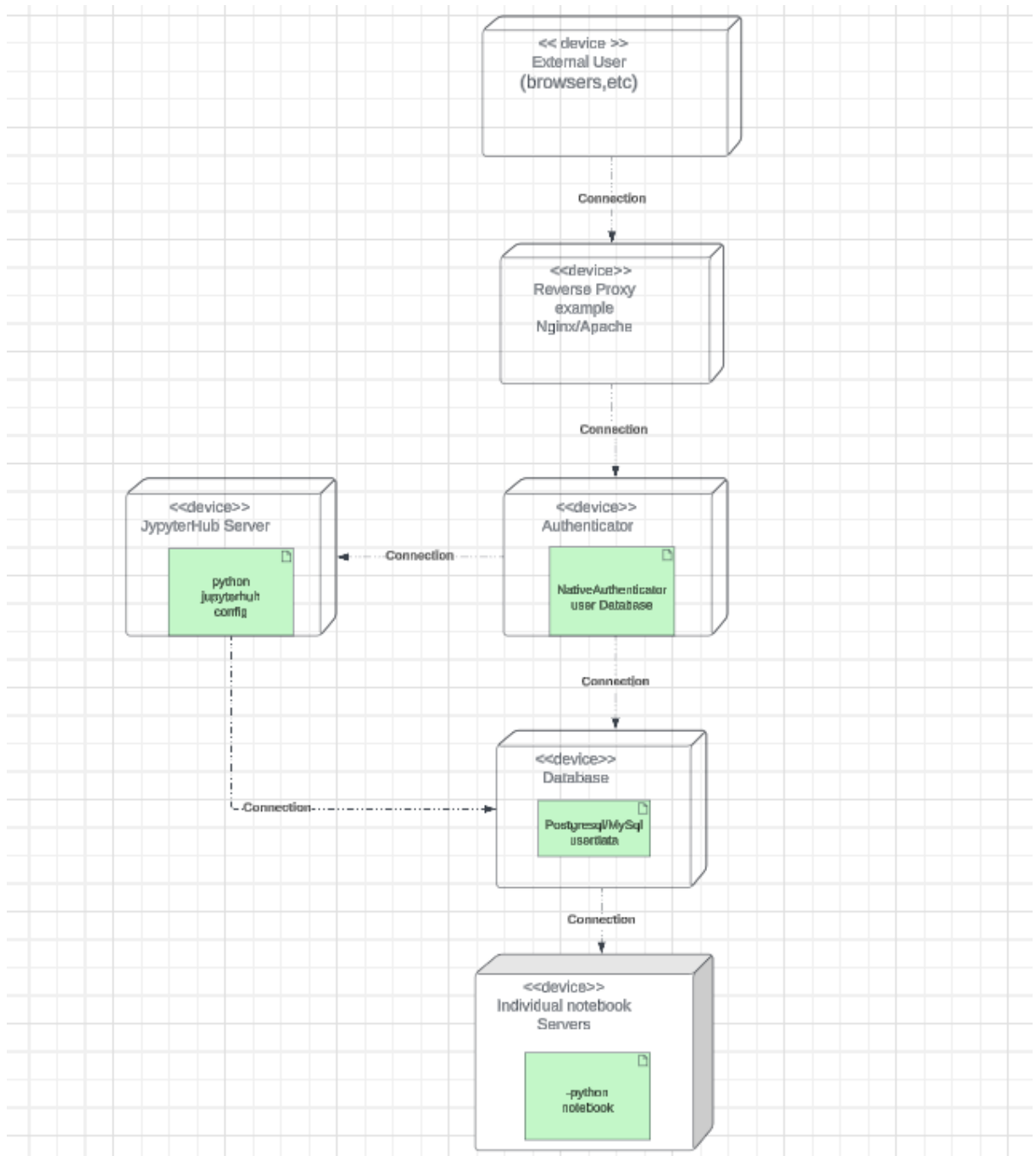
The above diagram is created through the Lucid app here is the link for a direct visit.

1. User Devices :- Devices used by users to access jupyterHub.
2. JupyterHub Server:- The server running jupyterHub, which includes the Authenticator, Proxy and spawner.
3. Database server:-The server hosting the database for user authentication and other data.

4. Notebook Servers:- individual jupyter Notebook servers spawned for each user.
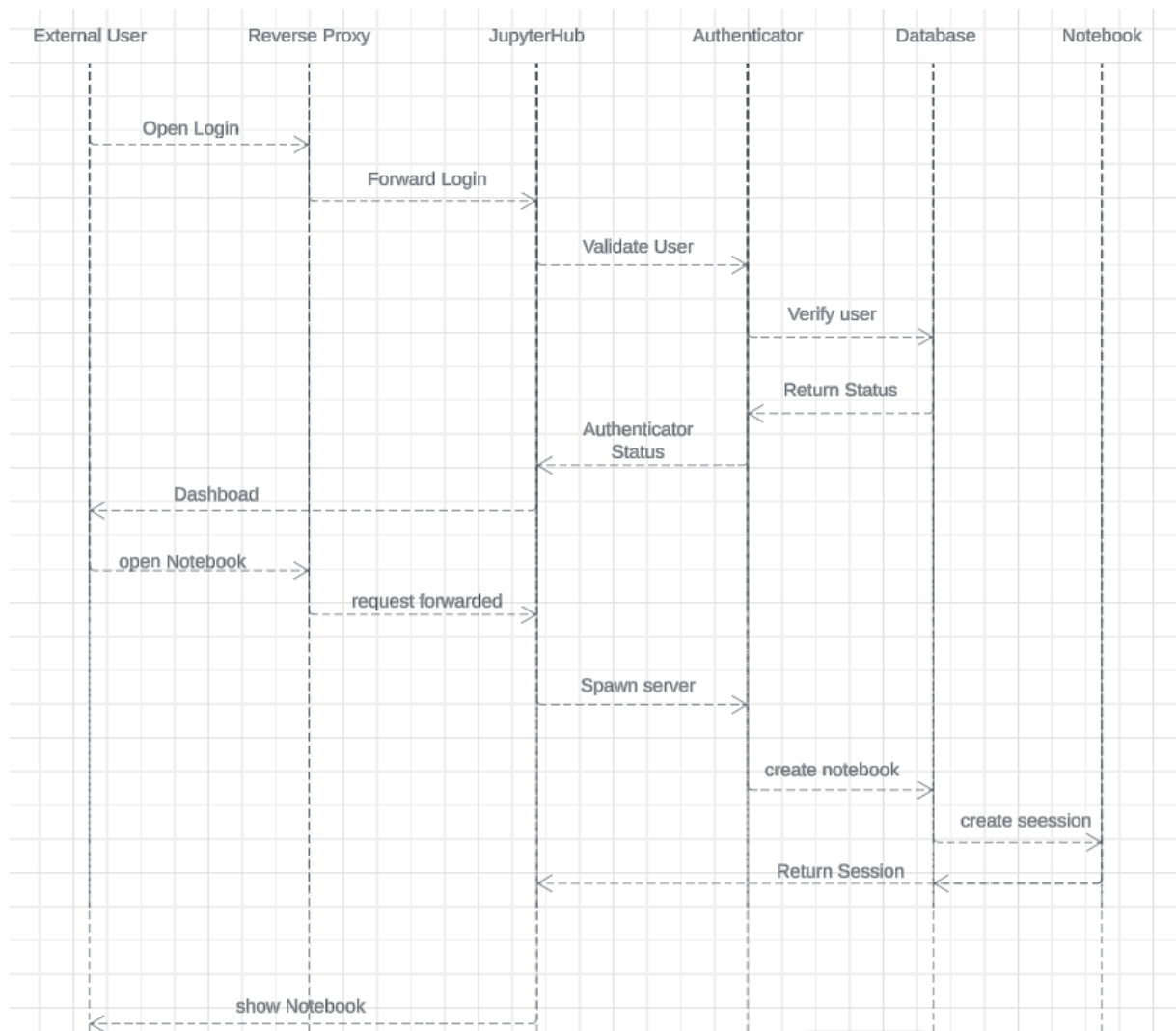
   Communication paths:

   User Devices <--------->jupyterHub Server: Users access jupyterHub through a web browser.

   jupyterHubServer<---------> Database Server: jupyterHub communicates with the database for user authentication and data storage.

   jupyterHubServer<--------->Notebook Servers: JupyterHub spawns and manages Notebook servers for each user.

## 6. Sequence Diagram

Sequence diagram

The above diagram is created through the Lucid app here is the link for a direct visit:

Explanation:-
1. From External User to Reverse Proxy: "Open Login "
2. From Reverse Proxy to jupyterHub:"Forward login"
3. From jupyterHub to Authenticator:"Validate User"
4. From Authenticator to Database:"verify User"
5. Database returns status to Authenticator.
6. Authenticator returns authentication to the External User.
7. External User requests to open a notebook.
8. Reverse Proxy forwards the request to jupyterHub.
9. jupyterHub spawns a server for the Notebook Server.
10. Notebook Server creates a new session and indicates readiness.
11. jupyterHub notifies the External User that the notebook is ready.

7. State Diagram
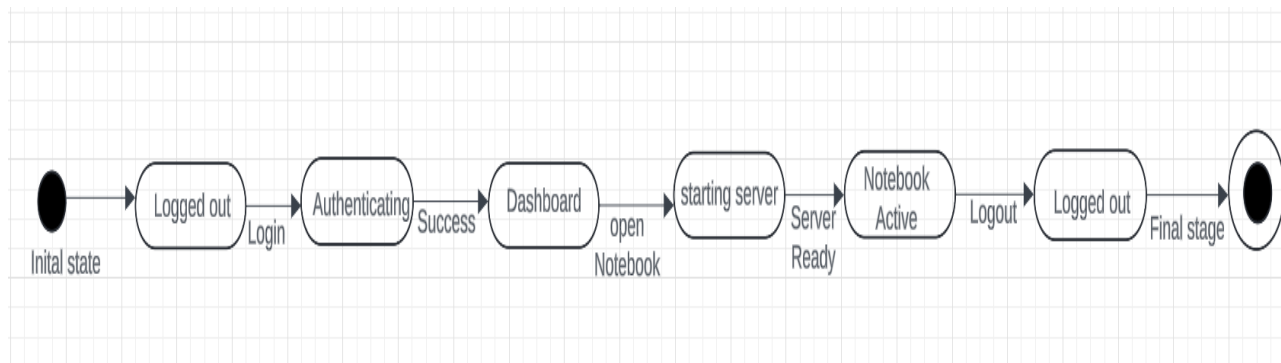
https://lucid.app/lucidchart/7b27baae-0c7a-484f-9283-b7141be4d936/edit?beaconFlowId=E912571B124BFF7D&invitationId=inv_b518ade4-b319-4051-90b1-416a060df2ca&page=0_0

The above diagram is created through the Lucid app here is the link for a direct visit:

8. BPMN Diagram

https://lucid.app/lucidchart/d44c0932-f432-4366-a260-07b0ef7558ac/edit?beaconFlowId=E070530E9FECA7E2&invitationId=inv_c6ab81ea-2fb5-4d50-99a8-0b600bc6b115&page=0_0

The above diagram is created through the Lucid app here is the link for a direct visit:

BPMN diagram:- Business Process Model and Notation are used to model a business process. For jupyter Notebook, explains the workflow of a user interacting with the notebook system.
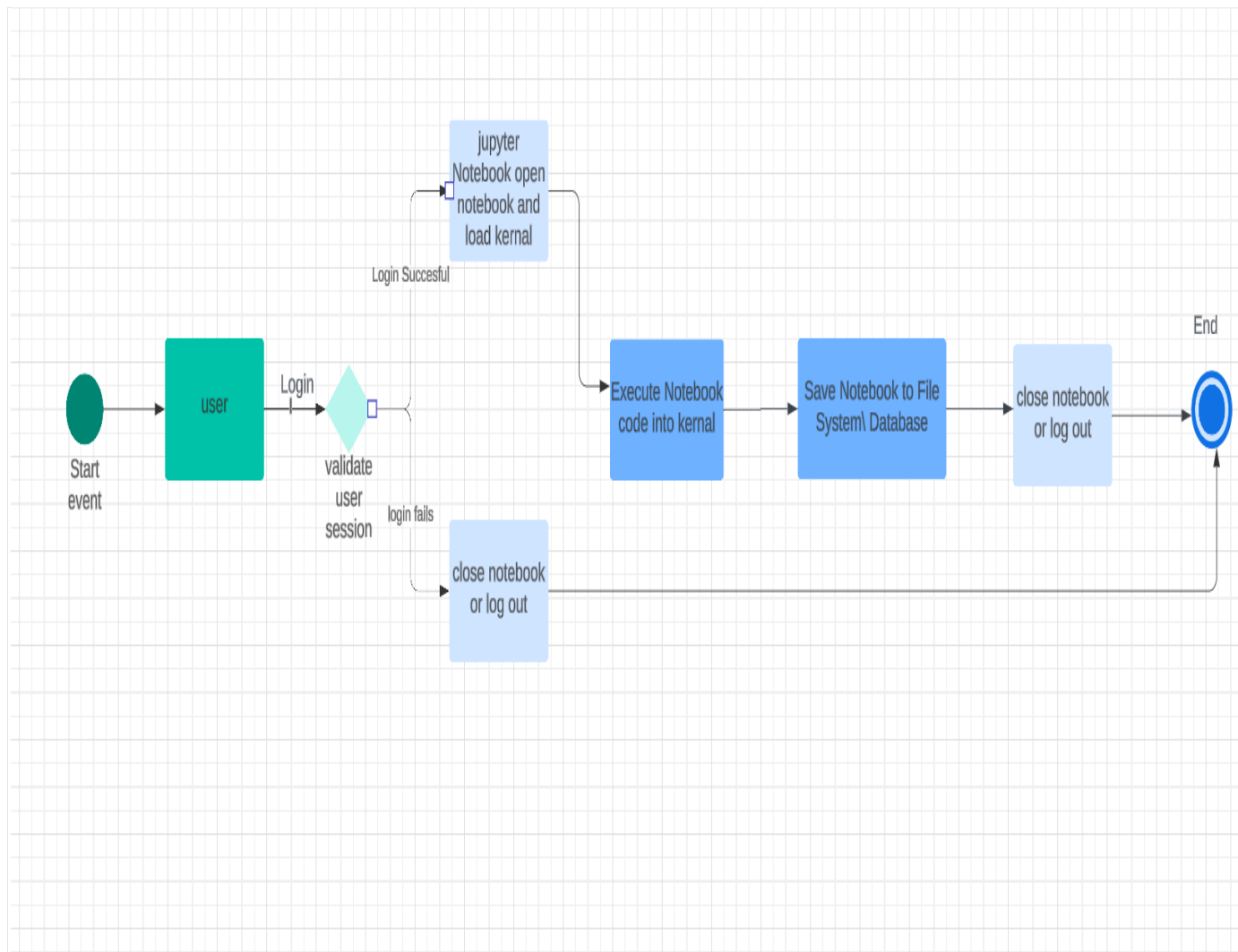
BPMN diagram:- Business process Model and Notation are used to model business process. For jupyter Notebook, explains the workflow of user interacting with the notebook system

1. User Interacts with Jupyter Notebook:
   The user logs in and opens a notebook.
   The system validates the user session.
2. Jupyter Notebook operations:
   The user interacts with the notebook, entering and running code.
   The notebook server manges code execution using Kernal.
3. Saving and Closing:
   The user saves the notebook.
   The notebook server saves the notebook data to the the file system or database.
   The user logs out and closes the notebook.

9. API modification for the current project is not required in my case because i am trying to acess the jupyter notebook locally using the command prompt and spawer opens it locally it is accessible to user through webbrowser.

References:-

Chatgpt

Google

Stackoverflow

Youtube video

Lucid app

Notes

Official document