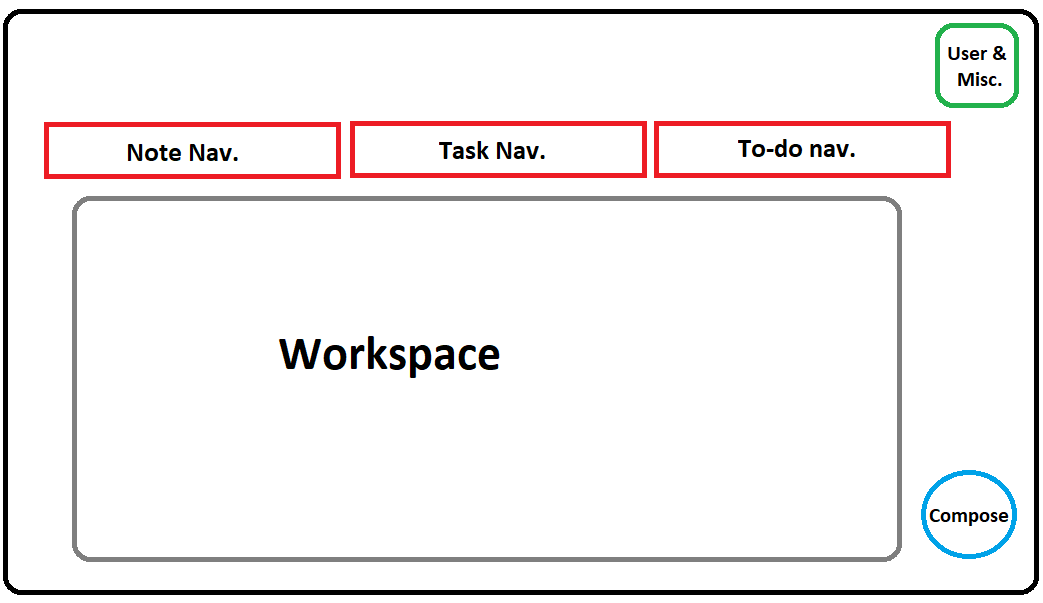
**3. Specific Requirements**

**3.1) External Interfaces**

User Interfaces



Description of each UI Buttons:-

* **Compose: To compose a new Note/Task/To-do based on current workspace selected.**
* **Note Navigation Button: To show current notes saved by user.**
* **Tasks Navigation Button: To show current tasks set by user.**
* **To-do Navigation Button: To show to-do tasks to be done by user.**
* **User & Miscellaneous- Button to display user details, Completion statistics, Settings, etc.**

Hardware Interfaces

* **OS: Windows 7 / 8/ 10 , Android**
* **Monitor: At least 1280x800 pixels in 256 colors**
* **A mouse or other pointing device & a keyboard**
* **Active internet connection**

Software Interfaces

* **OS: Windows 7 / 8/ 10**
* **IDE: Visual Studio Code**
* **Programming Language: PHP, JS, Python**
* **User Interface: HTML, CSS**
* **Server: WAMP**

Environment Interfaces

* **Visual Studio Code: IDE developed by Microsoft, used for making this web-app.**
* **WAMP server: PHP based local server for hosting the web-app.**
* **JS: JavaScript used for dynamic updation, controlling multimedia & UI.**
* **PHP: Hypertext Preprocessor used for developing the main back-end logic.**
* **Python: Python Programming used for mailing system.**
* **JSON: JavaScript Object Notation used for storing data.**
* **HTML: Hypertext Markup Language used for laying the basic foundation of the UI of the web-app.**
* **CSS: Cascading Style Sheet used for designing the elements rendered by HTML**.

**3.2) Functions**

The functional requirements of this web-application are:-

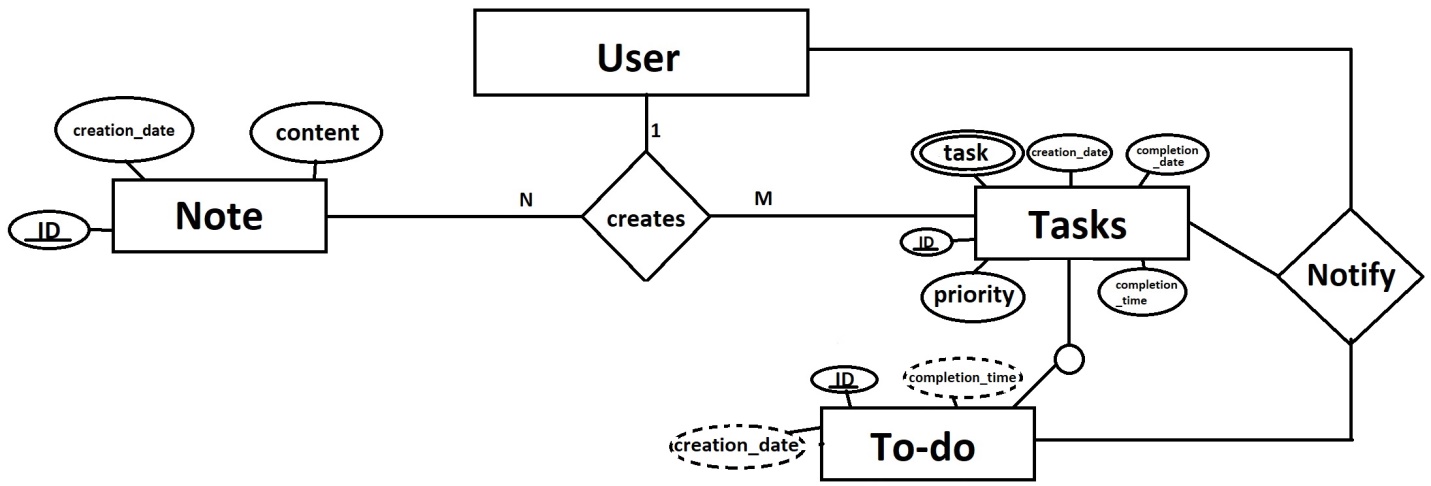
* **Register new users.**
* **Record the Notes/Tasks created by the user.**
* **Convert a task into a to-do list over a time-bound constraint.**
* **Generate notification alert of current tasks/to-do to be completed by the user.**
* **Generate various statistical reports based on productivity of the user.**

**3.3) Performance Requirements**

As of now, Noteffy has not undergone through a beta-testing, hence analyzing the current performance of the prototype is a dubious task.

**3.4) Logical Database Requirements**

EER (Enhanced Entity Relationship) Diagram

****

In this ER diagram, there are 4 entities: User, Note, Task & To-do.

The User entity is connected to the Note & Task entities through a "creates" relationship, which indicates that a user can create multiple notes & tasks.

The “To-do” entity is a subclass of the “Task” entity.

Task & To-do entities are connected back to the User entity through a “notify” relationship, which indicates that if a task or to-do is not completed by a completion\_time fixed by user, the user will be notified.

**3.5) Design Constraints**

**3.5.1) Standards Compliance**

* **‘Accessibility’ is an important idea behind many web standards, especially HTML. Not only does this mean allowing the web to be used by people with disabilities, but also allowing web pages to be understood by people using browsers other than the usual ones – including voice browsers that read web pages aloud to people with sight impairments, Braille browsers that translate text into Braille, hand-held browsers with very little monitor space, teletext displays, and other unusual output devices.**
* **On grounds of ‘Developer Agnostic’, Noteffy may go through several teams of designers during its lifetime, it is important that those people are able to comprehend the code & edit it easily. Complying to current Web standards, Noteffy should use open source softwares for its functioning & current versions of – HTML5, CSS3, PHP8, JS ES15, Python 3.10**
* **The graphical assets used for Noteffy are designed to be perceived with a quirky theme & must be made using latest version of Adobe Photoshop CC 2022.**

**3.6) Software System Attributes**

**3.6.1) Reliability**

As of now, the prototype of the web-app stores the note/task successfully created by the user through the interface. The back-end successfully notifies the user on his/her incomplete task.

On a small-scale, the reliability is well-intact.

**3.6.2) Availability**

The web-app will be available on the internet & anybody can access it via the internet.

The user data is well-intact on the server with full encryption.

In case a user is not able to complete his/her task, the incomplete task will be available in the miscellaneous tab of the web-app for the user’s reference- the user can access it any time.

If, any time in the future, the web-app is lost or gets corrupt due to obscure reasons, a backup of the app is well-kept & maintained.

**3.6.3) Security**

The user data is well-intact on the server & to avoid any case of external breach, the data is encrypted using Advanced Encryption Standard (.aes).

**3.6.4) Maintainability**

**3.6.5) Portability**

**3.7) Organizing the Specific Requirements**

**3.7.1) Functional Hierarchy**