**2. General Requirements**

**2.1) Product Perspective**

2.1.1)System Interfaces :

* **Noteffy is a completely self contained software that does not belong to any larger ecosystem of products**

2.1.2)User Interfaces :

* **Users interface with Noteffy through a website:**

-New users register their details through the authentication pagedescribed in section 3.1 and visualized in fig.(1)

-Recurring users register their details through a personal dashboard page described in section 3.1 and visualized in fig.(2)

-Recurring users can view their monthly productivity charts and other statistics through a personal scoreboard page described in section 3.1 and visualized in fig(3)

-Recurring users(with Admin priveleges) can view and manage their joined and created workspaces through an admin panel described in section 3.1 and visualized in fig(5)

-Recurring users(with Admin privileges) can view their monthly productivity charts,assigned schedules and class members' statistics through a personal scoreboard page described in section 3.1 and visualized in fig(6)

2.1.3)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**

2.1.4) Software 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 (ES13) used for dynamic updation, controlling multimedia, UI and API communication.The libraries used were the standard library with JSON and fetch support and the chart API.**
* **PHP: Hypertext Preprocessor used for developing the main back-end logic. Only the standard library was used.**
* **Python: Python Programming languages used for mailing system. Libraries used will be the standard library and smtplib(for mailing).**
* **JSON: JavaScript Object Notation used for storing data.**
* **HTML5: Hypertext Markup Language used for laying the basic foundation of the UI of the web-app.**
* **CSS6: Cascading Style Sheet used for styling the elements rendered by HTML**.

2.1.5) Communication Interfaces

* **To implement the notification functionality, the mailing API in python will use the Gmail SMTP relay server through the smtplib library.**

2.1.6) Memory constraints

* **No such constraint**

2.1.7) Operations

* **There are 5 modes of operation for the user:**

**-**New users(with no cached data) interact through authentication page

-Returning users(with cached data) interact with the dashboard and create,delete and edit the notes and tasks that they make.

-Returning users also get to complete and edit to-do lists and view their monthly productivity charts.

-Returning users(with Admin privileges) get to create workspaces/domains where other users can enroll, can assign tasks to them and schedule events for a workspace and appositely can also join other workspaces and recieve the tasks assigned within that workspace.

-Returning users (with Admin privileges) also get to review the performance of their workspace members using control charts that show the user's monthly productivity.

2.1.8) Site adaptation requirements

* **No such constraint**

**2.1) Product 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 charts based on productivity of the user.**
* **Allow users to gain admin privilieges**
* **Allow admin users to create and join workspaces**
* **Allow admin users to assign tasks and schedule tasks in the workspaces that they have created**
* **Also, allow admin users to assess the productivity of various workspace members via generated statistical charts**
* **Generate notification alerts for various admin assigned tasks and events given in a workspace**
* **Allow admin users to complete tasks and recieve events in the workspaces that they have joined.**

**2.3) User characteristics**

The ideal users for the web-application are:-

* **Casual users looking to increase their productivity and have better time management**
* **Professionals seeking to better schedule their workload and keep track of it across platforms (mobile phone,workstations,etc.)**
* **Professionals seeking to organize and manage teams by tracking their workload, planning events and assesssing their productivity via easy-to-understand, intuitive interfaces**
* **Young to medium aged users who want to socially interact and positively compete with their friends,colleagues or acquantances online on the basis of productivity**

**2.4) Constraints**

* **The product shall meet accessibility standards to make the application inclusive for all potential user groups and the developer agnosticism to make the development process robust and sustainable.**
* **The user database has to be securely encrypted to prevent data breach and violation of personal privacy.**
* **System criticality is medium level as only email id is provided by the user that can be misused but safety parameters have still been put in place**

**2.5) Assumptions and Dependencies**

* **The user database has to be securely encrypted to prevent data breach and violation of personal privacy.**
* **The application interacts with cronjob api which is only compatible with Linux operating system. For windows OS, correlating api would be task scheduler**

**2.6) Apportioning of Requirements**

* **The quirky theme has to be added to workspace and dashboard segments in order the maintain UI consistency.**
* **OTP functionality has to be reorganized to ensure user data security**
* **Developmental administrative panel modules have to be constructed to store expired tasks and provide control and maintenance functions.**
* **The web application is partially responsive and so behaves visually as intended on desktop environments.**