ASSIGNMENT-1

SRS Report And Contextual Enquiry

Group Members:

Aditya Rajesh Patil	**180101004**
Kartikeya Saxena	**180101034**
Milind Prabhu	**180101091**
Kushal Sangwan	**180101096**
Sudesh Chaudhary	**180103077**

Problem Statement

Problem 2: Class Note taking application:
Taking good notes in class is an integral part of academic success in college. Actively taking notes during class sessions can help you focus and better understand the main concepts. Taking notes on both synchronous and asynchronous material can help students better remember what they hear and see. Efficient and concise notes can also save time, energy, and confusion for a student. There is no right format to use when taking notes. Preferably, many different structures and styles can be used to take quality notes. The following way a student can do the note-taking :

1. Keeping sections for the date, essential question, topic, notes, questions, and a summary; using flow charts and concept maps;
2. Writing outline to organize the lecture by main points, allowing room for examples and details.
3. Using a flowchart/concept map to represent the lecture content visually
4. Organizing notes from lectures with a substantial amount of facts through dividing key topics into columns and recording facts underneath
5. Or quickly covering essential details and information. Design and develop software that can be useful for a student while taking class notes.

Purpose of the Document

The purpose of this document is to build a note taking application to aid students to take notes with ease and organize them effectively. The document will exhaustively list the features of the software and will be presented to a customer to seek their approval. It will also serve as a reference to the developers building the first version of the software.

Scope of the System and Target Audience

The Notes taking application is intended to be used by students in the classroom environment. The application is expected to help them organise and write notes both synchronously and asynchronously with ease. It should allow the students to improve their efficiency in taking notes, giving them more time to focus on the lectures. Moreover, it is also expected to provide the professors and teachers with a platform to share their handouts with the students.

Overview Of The Document

This SRS Document is divided into two parts to provide the user with functional and nonfunctional requirements of the Notes taking application system. The first section deals with gathering of various functional requirements and their hierarchy as concluded by an iterative process of speculating and contextual enquiries. Furthermore, the second section deals with the gathering of non-functional requirements particularly “Usability” through Active Contextual Inquiry and representation of our collective understanding of the user through Affinity Diagram.

Functional Requirements Hierarchy

F.1. Create Account
F.2. Sign in
F.3. Sign out
F.4. Delete Account

F.5. Create New Subject
F.6. Rename Subject
F.7. Delete Subject
F.8. Manage Subject
F.8.1. Create New Topic
F.8.2. Rename Topic
F.8.3. Delete Topic
F.8.4. Manage Topic
F.8.4.1. Create New Note
F.8.4.2. Delete Note
F.8.4.3. Rename Note
F.8.4.4. Manage Note

F.8.4.4.1. Insert Text
F.8.4.4.2. Edit Text
F.8.4.4.2.1. Change font size
F.8.4.4.2.2. Change font face
F.8.4.4.2.3. Change font style
F.8.4.4.2.4. Change text color
F.8.4.4.2.5. Highlight Text
F.8.4.4.3. Delete Text

F.8.4.4.4. Insert Equation
F.8.4.4.5. Delete Equation

F.8.4.4.6. Insert Hyperlink
F.8.4.4.7. Delete Hyperlink

F.8.4.4.8. Insert Image
F.8.4.4.9. Delete Image

F.8.4.4.10. Insert Flowchart
F.8.4.4.11. Delete Flowchart

F.8.4.4.12. Insert Handwritten Stroke
F.8.4.4.13. Delete Handwritten Stroke

F.8.4.4.14. Insert Bookmark
F.8.4.4.15. Delete Bookmark

F.8.4.4.16. Undo Change
F.8.4.4.17. Redo Change

F.9. Search Note by Title Name
F.10. Search Note by Keywords
F.11. Sort by Modification Time
F.12. Sort by Name

F.13. Add Collaborator
F.14. Accept Collaboration
F.15. Export Note
F.16. Make To-do List
F.16.1. Add Task
F.16.2. Tick Task
F.16.3. Untick Task
F.16.4. Delete Task

Functional Requirements

Authentication Requirements

R.1. Create Account
Input: Credentials
Output: Account ID
Description: Registers a new account if valid credentials and returns a unique account ID otherwise outputs error message.

R.2. Sign in
Input: Credentials
Output: Success/Error message
Description: Signs the user into the account if it exists otherwise outputs error message.

R.3. Sign out
Input: Account ID
Output: Success/Error message
Description: Signs the user out of the account if previously logged in otherwise throws an error message.

R.4. Delete Account
Input: Account ID
Output: Success/Error message
Description: A user can delete their account using this function if the account exists.

Modification Requirements

R.5. Create New Subject
Input: Subject name
Output: Subject folders list
Description: Creates a new subject folder with given Subject name and adds it to the subject folders list if it does not exist already. Otherwise outputs a duplicate error message.

R.6. Rename Subject
Input: Subject name 1, Subject name 2
Output: Subject folders list
Description: Changes the subject name of the subject folder with the Subject name 1 to Subject Name 2. Throws not found error if Subject name 1 not found or duplicate error if Subject Name 2 already exists.

R.7. Delete Subject
Input: Subject name
Output: Subject folders list

Description: Deletes subject folder with the given subject name from the subject folders list if it exists otherwise shows not found error

R.8. Manage Subject

Description: Makes changes to an existing subject using functions described below.

R.8.1. Create New Topic

Input: Topic name
Output: Topic folders list
Description: Several Topics can be created under a Subject. It appends a new topic folder with the given Topic name to the topic list if it does not exist already otherwise outputs a duplicate error message .

R.8.2. Rename Topic

Input: Topic name 1, Topic Name 2
Output: Topic folders list
Description: Changes the topic name of the topic folder with the Topic name 1 to Topic Name 2. Throws not found error if Topic name 1 not found or duplicate error if Topic Name 2 already exists.

R.8.3. Delete Topic

Input: Topic name
Output: Topic folders list
Description: Deletes topic with the given Topic name from Topic folders list if topic name already exists otherwise throws not found error.

R.8.4. Manage Topic

Description: Makes changes to an existing topic using functions described below.

R.8.4.1. Create New Note

Input: Note title
Output: Notes list
Description: A Topic folder consists of several notes. It appends a new note with the given note title to the notes list if it does not exist already otherwise outputs a duplicate error message .

R.8.4.2. Delete Note

Input: Note title
Output: Notes list
Description: Deletes note with the given note title from notes list if it already exists otherwise throws not found error.

R.8.4.3. Rename Note

Input: Note title 1, Note title 2
Output: Notes list
Description: Changes the Note title of the note with Note title 1 to Note title 2. Throws not found error if Note title 1 not found or duplicate error if Note title 2 already exists.

R.8.4.4. Manage Note

Description: Makes changes to an existing note using functions described below.

R.8.4.4.1. Insert Text

Input: Text, Position
Output: Note
Description: Inserts the given text at the given position in the note and returns the updated note.

R.8.4.4.2. Edit Text

Description: Makes changes to an existing text using functions described below.

R.8.4.4.2.1. Change font size

Input: Text location, Size
Output: Note
Description: Increases the size of the text at the given text location to the given size and returns the updated note.

R.8.4.4.2.2. Change font face

Input: Text location, Font face
Output: Note
Description: Changes the font face of the text at the given location to the given font face and returns the updated note.

R.8.4.4.2.3. Change font style

Input: Text location, font style
Output: Note
Description: Changes the font style of the text at the given location to the font style and returns the updated note. The font style could be bold, italic, underline or strikethrough.

R.8.4.4.2.4. Change text color

Input: Text location, Color
Output: Note
Description: Changes the color of the text at the given location to the input color.

R.8.4.4.2.5. Highlight Text

Input: Text location, Color
Output: Note
Description: Highlights the text at the given location with the given color.

R.8.4.4.3. Delete Text

Input: Text location
Output: Note
Description: Deletes the text at the given location and returns the updated note.

R.8.4.4.4. Insert Equation
Input: Equation, Position
Output: Note
Description: Inserts the given equation at the given position in the note and returns the updated note.

R.8.4.4.5. Delete Equation
Input: Note location
Output: Note
Description: Deletes the equation at the given location and returns the updated note.

R.8.4.4.6. Insert Hyperlink
Input: Hyperlink, position
Output: Note
Description: Inserts the given hyperlink at the given position in the note and returns the updated note.

R.8.4.4.7. Delete Hyperlink
Input: Note location
Output: Note
Description: Deletes the hyperlink at the given location from the note

R.8.4.4.8. Insert Image
Input: Image, position
Output: Note
Description: Inserts the given image at the given position in the note and returns the updated note.

R.8.4.4.9. Delete Image
Input: Note location
Output: Note
Description: Deletes the image at the given location from the note

R.8.4.4.10. Insert Flowchart
Input: Flowchart, position
Output: Note
Description: Inserts the given flowchart at the given position in the note and returns the updated note.

R.8.4.4.11. Delete Flowchart
Input: Note location
Output: Note
Description: Deletes the flowchart at the given location from the note

R.8.4.4.12. Insert Hand Stroke
Input: Hand Stroke, position
Output: Note
Description: Inserts the given stroke at the given position in the note and returns the updated note.

R.8.4.4.13. Delete Hand Stroke
Input: Note location
Output: Note
Description: Deletes the handwritten stroke at the given location from the note

R.8.4.4.14. Insert Bookmark
Input: Bookmark title, position
Output: Note
Description: Inserts a bookmark with the given title at the given position in the note and returns the updated note. If a bookmark with the same title exists it throws duplicate error.

R.8.4.4.15. Delete Bookmark
Input: Bookmark title
Output: Success/Fail
Description: Deletes the bookmark with the given title from the note. If such a bookmark does not exist it outputs fail.

R.8.4.4.16. Undo Change
Input: Note title
Output: Note
Description: Undoes the last modification performed by the user the given note.

R.8.4.4.17. Redo Change
Input: Note title.
Output: Note
Description: It restores the last modification that was undone by the user in the given note. If the user never used undo it does not modify the note.

Searching And Sorting Requirements

R.9. Search Note by Title Name
Input: Note title
Output: Notes list

Description: Goes through all notes in a topic folder and returns a list of notes whose title substring match the given title.

R.10. Search Note by Keywords
Input: list of keywords
Output: Notes List
Description: Locates all occurrences of keywords inside all notes in a topic folder and returns a list of notes with these occurrences.

R.11. Sort by Modification Time
Input: Notes list, order flag
Output: Notes list
Description: Arranges all notes inside a topic folder according to decreasing/increasing order of their modification time based on the order flag.

R.12. Sort by Name
Input: Notes, list, order flag
Output: Notes list
Description: Arranges all notes inside a topic folder according to ascending/descending lexicographical order of their title based on the order flag.

Sharing Requirements

R.13. Add Collaborator
Input: Account ID, note title
Output: Invite Link
Description: Sends invitation link to another user with the given Account ID for the edit access to the note with the given title.

R.14. Accept Collaboration
Input: Link
Output: Note
Description: Provides the user edit access to the note in the Link.

R.15. Export Note
Input: Note title, format
Output: Note
Description: Converts a copy of the note with the given Note title to given format and returns the formatted note

Miscellaneous Requirements

R.16. Narrate Note:
Input: Note
Output: Voice
Description: narrates text written in the notes to the user

R.17 Privacy of Notes

Input: user1 update, user2 update
Output: note
Description: display same note for both users by incorporating both changes

R.18. To do-list

Input: Tasks
Output: To do list in note
Description: makes an ordered list of the tasks the user needs to do.

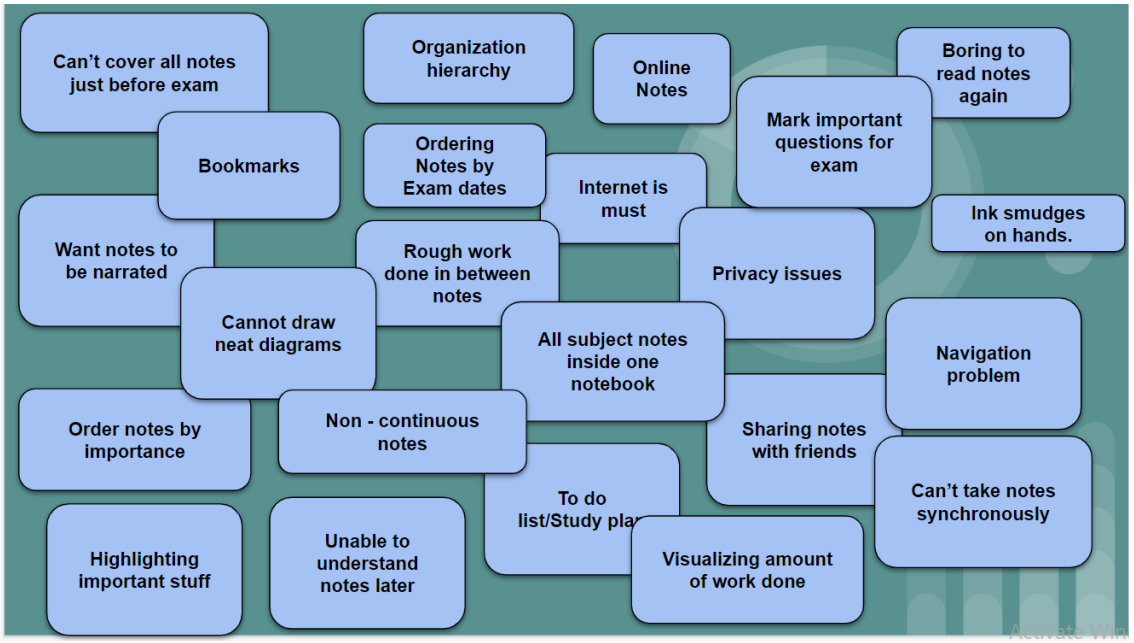
Contextual Enquiry

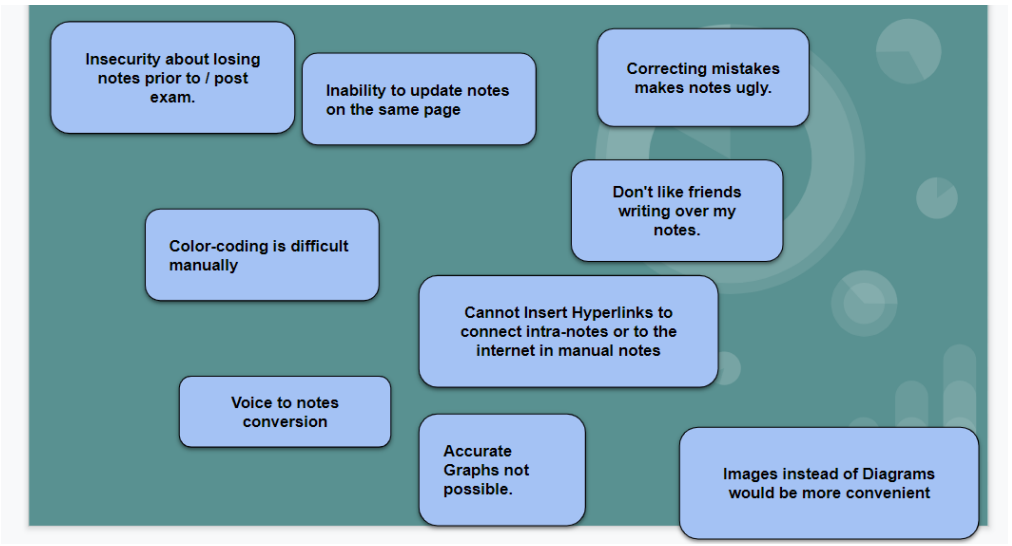
METHOD OF CONDUCTING:

As observing users in their natural work environment is necessary for a contextual enquiry , all five members of the group watched one or two friends of theirs while they took notes both attending live lectures(active observation) or listening to recorded lectures. Users were observed to see how they write their notes daily , arrange them , the problems they face while writing them and what are the inefficiencies which can be solved. The following thought ensembles were gathered from them , which were arranged into modules later in the affinity diagram.

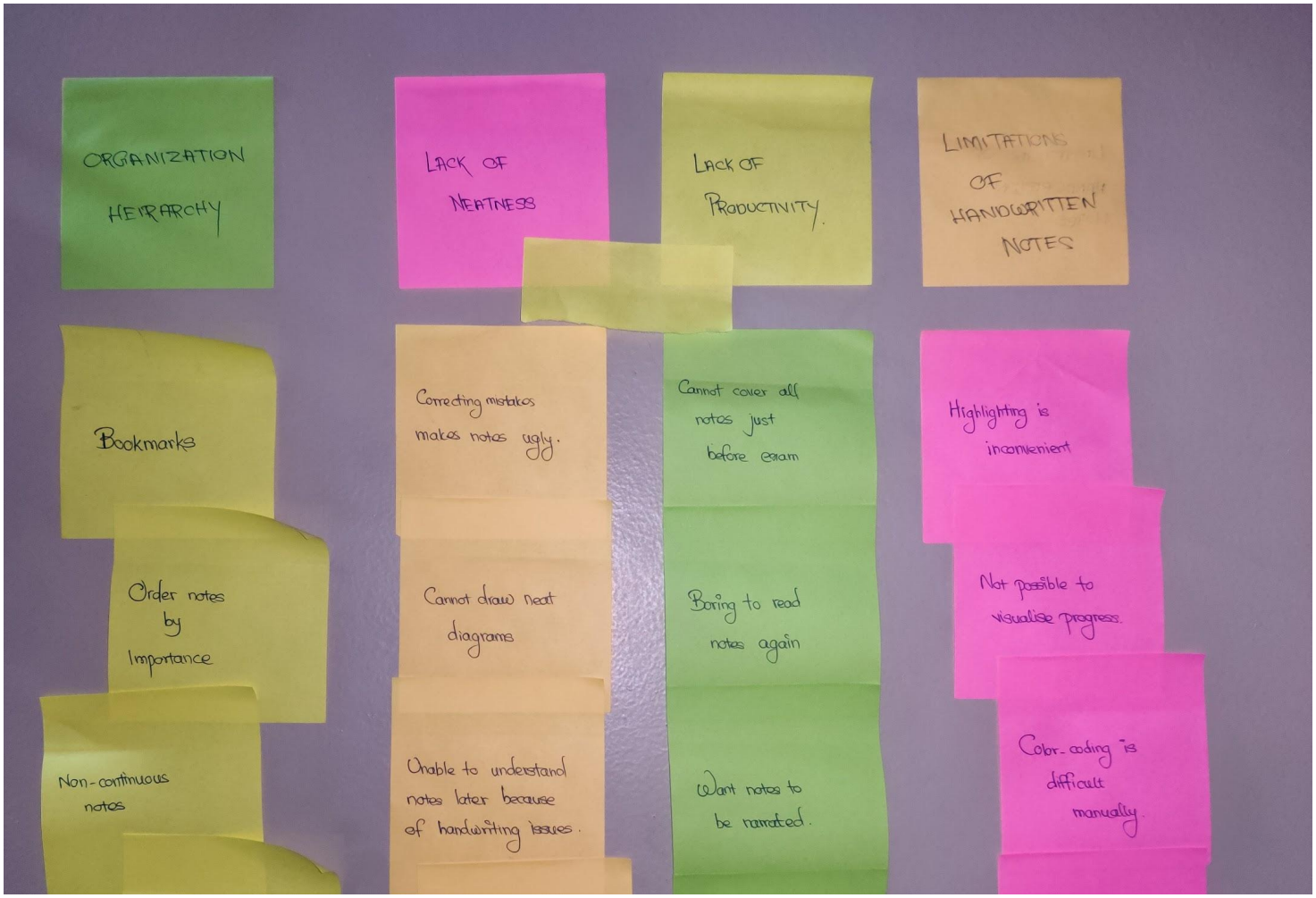
Its results were further used to define requirements, improve the process and learn what is important to users and customers

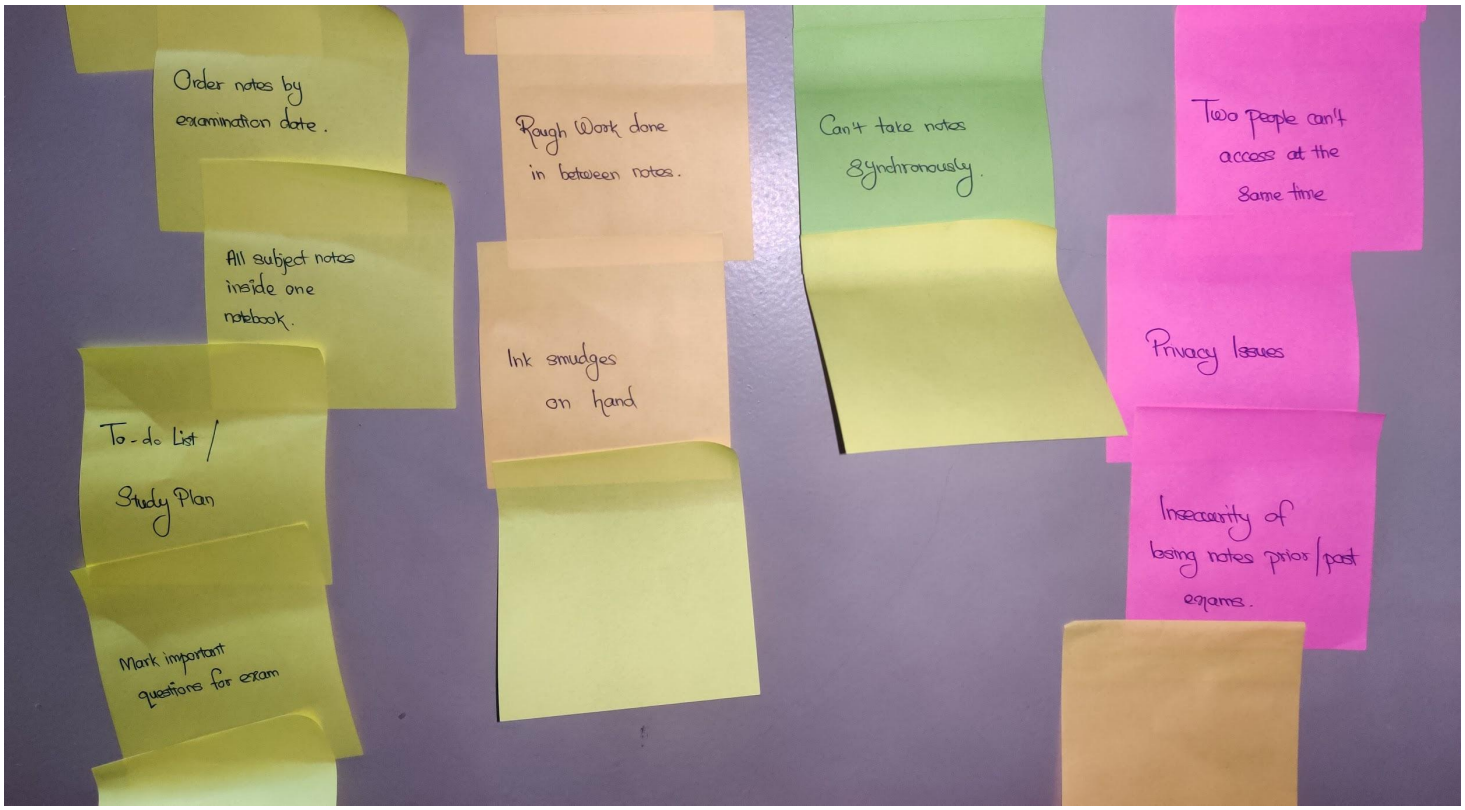
Initial Observations Gathered

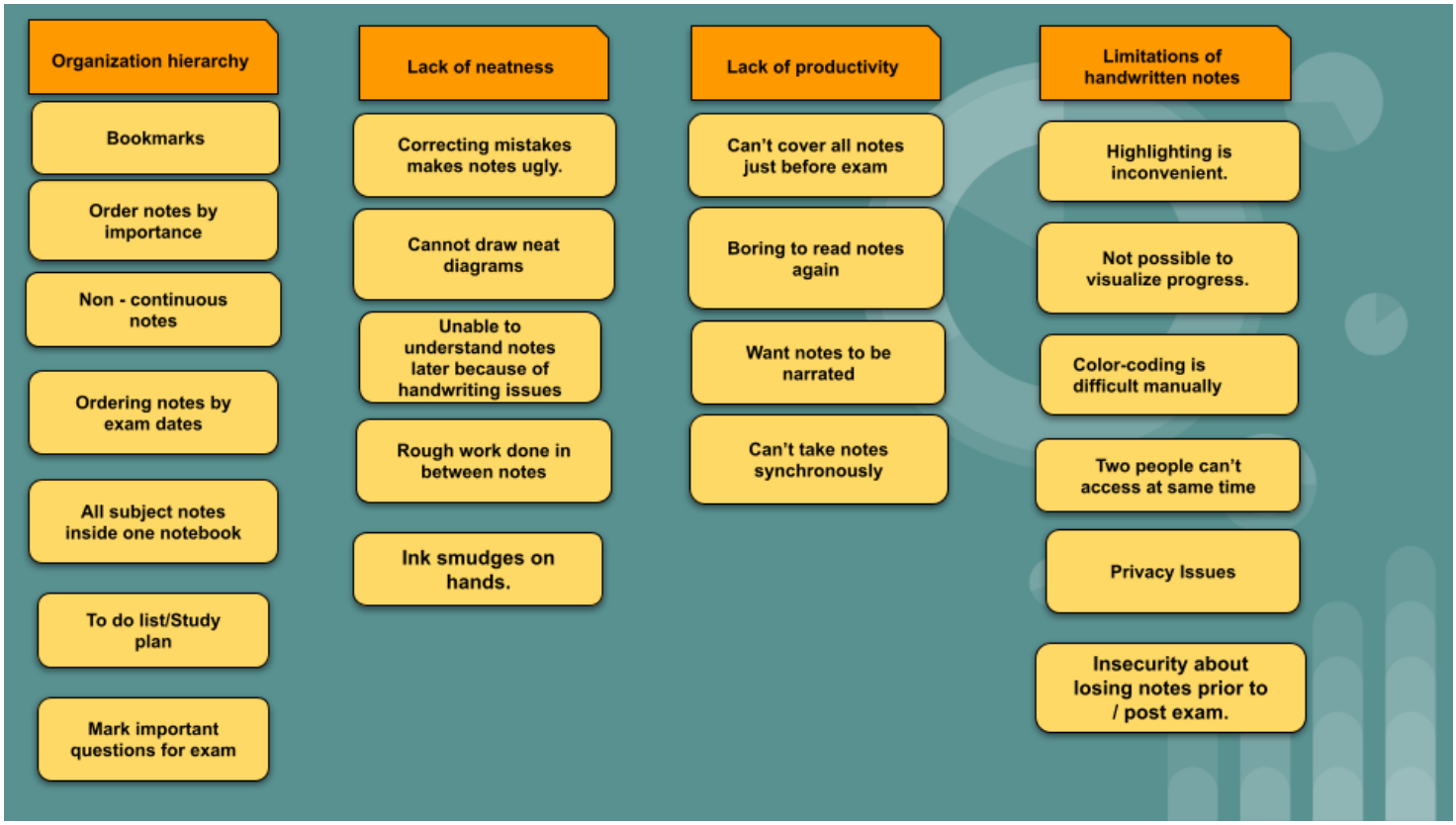




Affinity Diagram







Functional requirements in hierarchy derived from affinity diagram

1.Bookmarks - F.8.4.4.14
Users wanted to bookmark important stuff so that they can access it later easily.

2.Ordering Notes by Dates - F.11
As sometimes users accessed notes by the time or date when they wrote them in order to understand what they are currently studying better , ordering by modification time was created for this functionality

3.All Subjects in One place - F.5
As users sometimes wrote different subject notes in same place as a result making them unorganized , the create subject function was create to solve this problem by separating notes for different subjects

4. To-do List - F.16
Users frequently created a to- do list while studying to arrange the topics or subjects according to when they want to start them.

5. Cannot draw neat diagrams - F.8.4.4.8
Users sometimes needed to draw flow charts in order to understand concepts better , but drawing a neat and organized one was a tough task to do , hence inserting flow chart function was added.

6.Highlighting facts- F.8.4.4.2.5

Most users highlighted important topics/questions/concepts and hence highlight text functionality was included

7.Color Coding - F.8.4.4.2.4
Color coding the same type of concepts / sections helped users in visualizing their work.

8.Two People Accessing at same time - F.13 - F.15
Two users many a times needed to share notes in order to understand concepts better , hence note collaboration was added

9.Insecurity of Losing Notes - Cloud Compatibility
Users misplace notes before exams. Some even want to keep them after exams for future reference but eventually lose them. Storing notes in Cloud eliminates the possibility of misplacing notes.

10.Ink Smudges on hand - Stylus Compatibility
Users wanted the handwriting experience without all its problems. Making the app compatible with the stylus appealed to them.