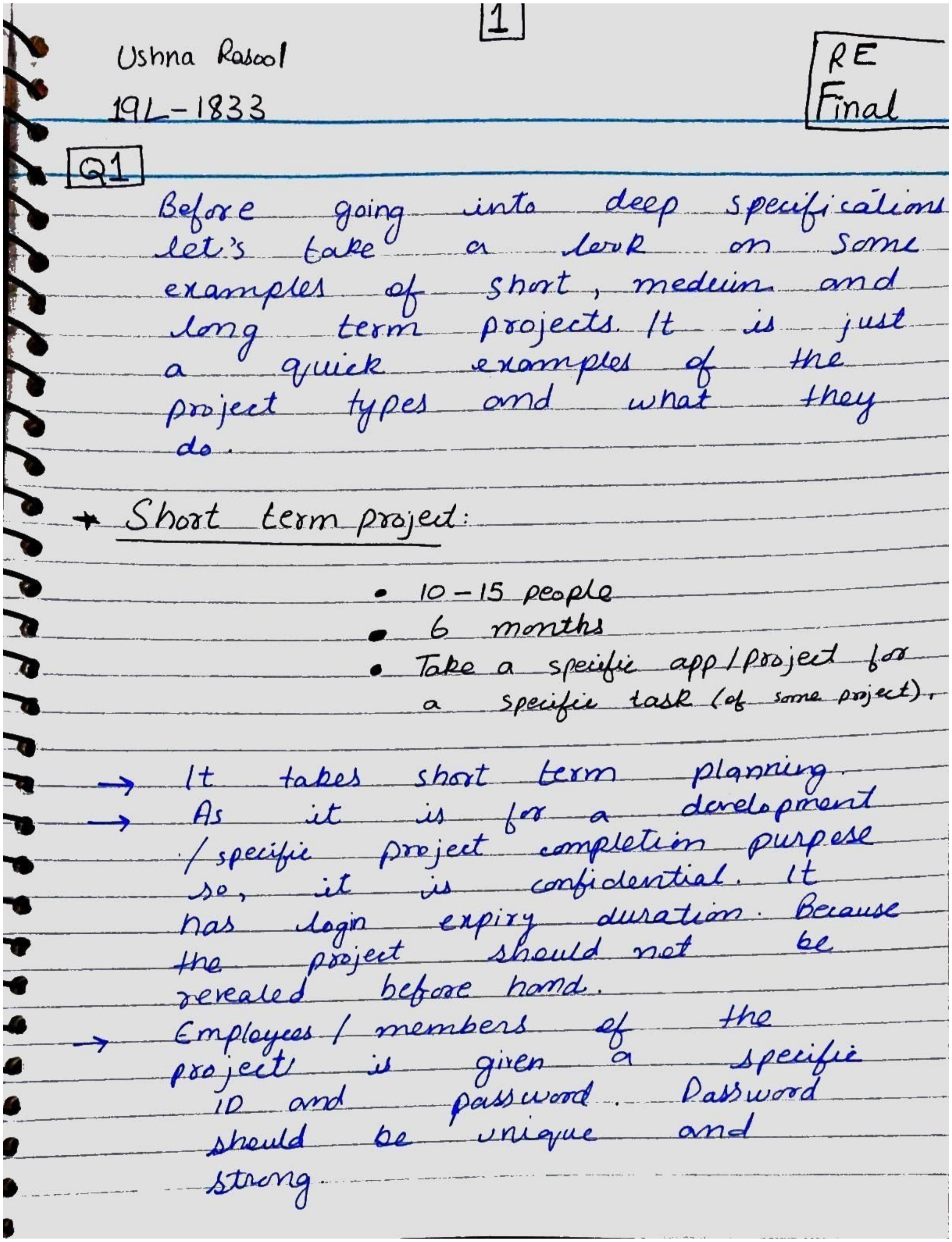
RE FINAL TERM

Presented to

Sir Umair

Presented by

Ushna rasool
19L-1833



1

Ushna Rasool

19L-1833

RE
Final

Q1

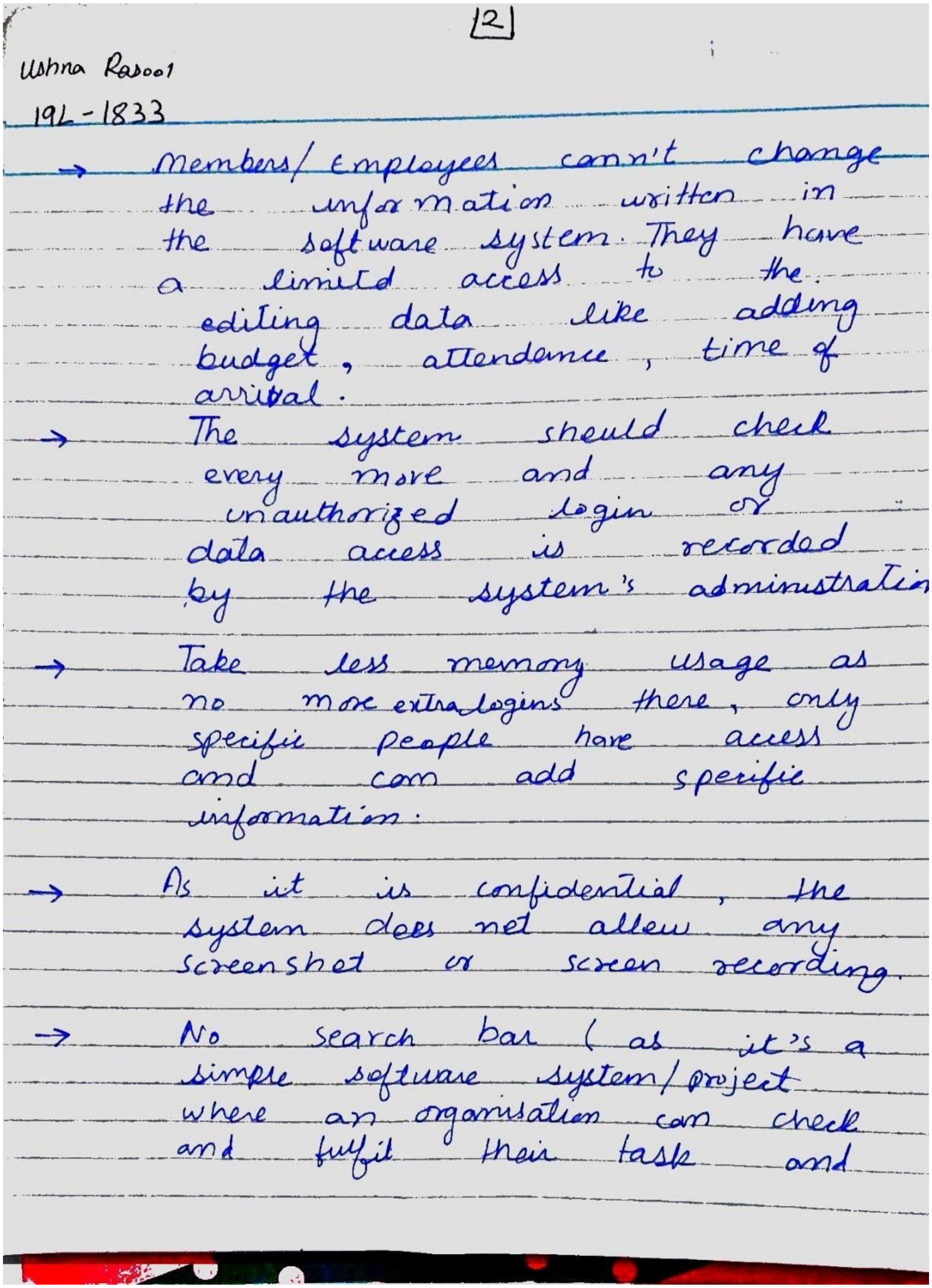
Before going into deep specifications let's take a look on some examples of short, medium and long term projects. It is just a quick examples of the project types and what they do.

+ Short term project:

- 10 – 15 people
- 6 months
- Take a specific app / project for a specific task (of some project).

→ It takes short term planning.
→ As it is for a development / specific project completion purpose so, it is confidential. It has login expiry duration. Because the project should not be revealed before hand.
→ Employees / members of the project is given a specific ID and password. Password should be unique and strong.

USHNA RASOOL
19L-1833



Ushna Rasool
19L-1833

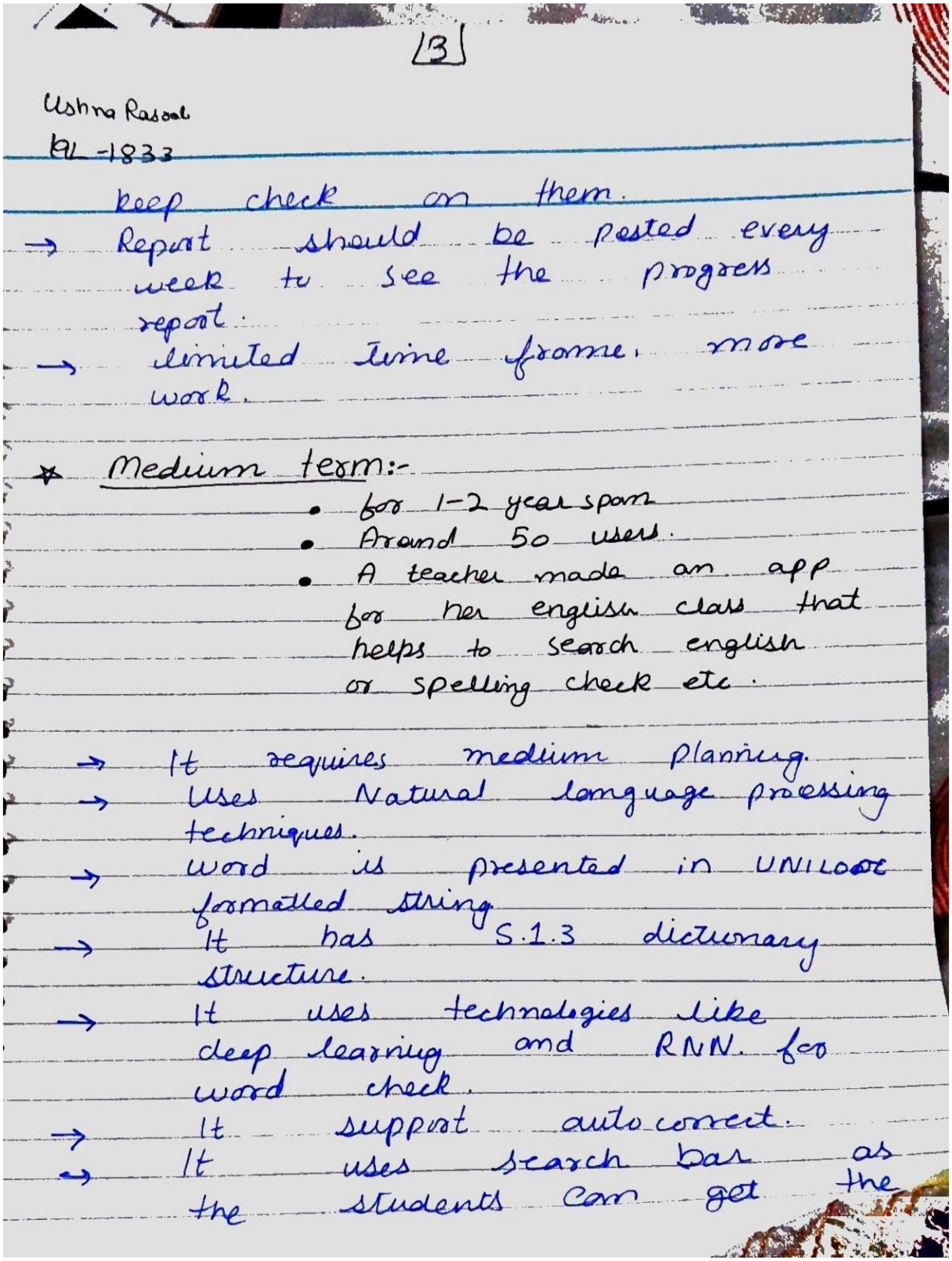
→ Members/Employees cann't change the information written in the software system. They have a limited access to the editing data like adding budget, attendence, time of arrival.

→ The system should check every move and any unauthorized login or data access is recorded by the system's administration

→ Take less memory usage as no more extra logins there, only specific people have access and can add specific information.

→ As it is confidential, the system does not allow any screenshot or screen recording.

→ No search bar (as it's a simple software system/project where an organisation can check and fulfil their task and



[3]

Ushna Rasool
19L-1833

keep check on them.

→ Report should be posted every week to see the progress report.

→ limited time frame. more work.

★ <u>Medium term:-</u>

- for 1-2 year span
- Around 50 users.
- A teacher made an app for her english class that helps to search english or spelling check etc.

→ It requires medium planning.

→ Uses Natural language processing techniques.

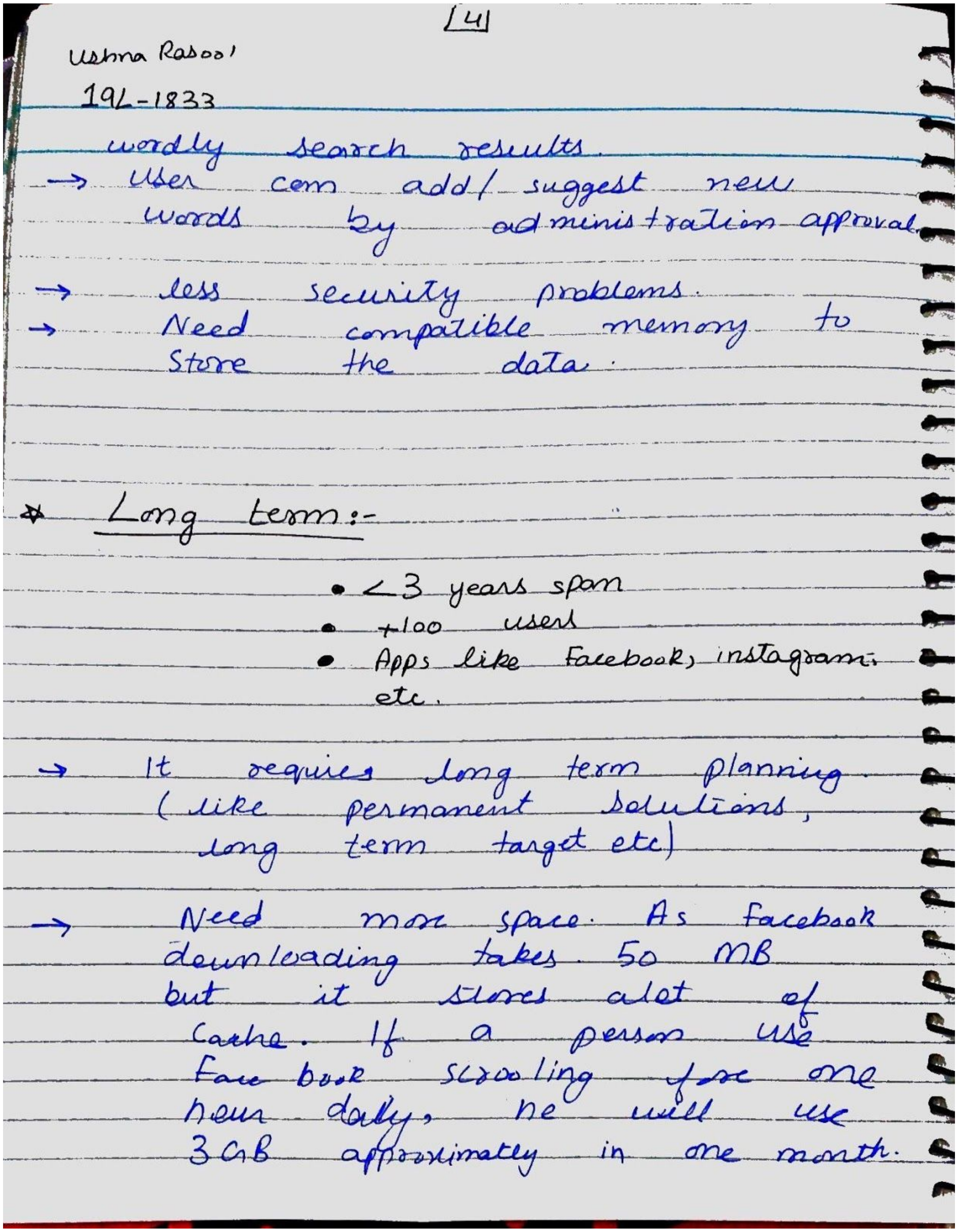
→ word is presented in UNICODE formatted string

→ It has S.1.3 dictionary structure.

→ It uses technologies like deep learning and RNN. for word check.

→ It support autocorrect.

→ It uses search bar as the students can get the



[4]

Ushna Rasool

19L-1833

wordly search results.

→ User can add/ suggest new words by administration approval.

→ less security problems.

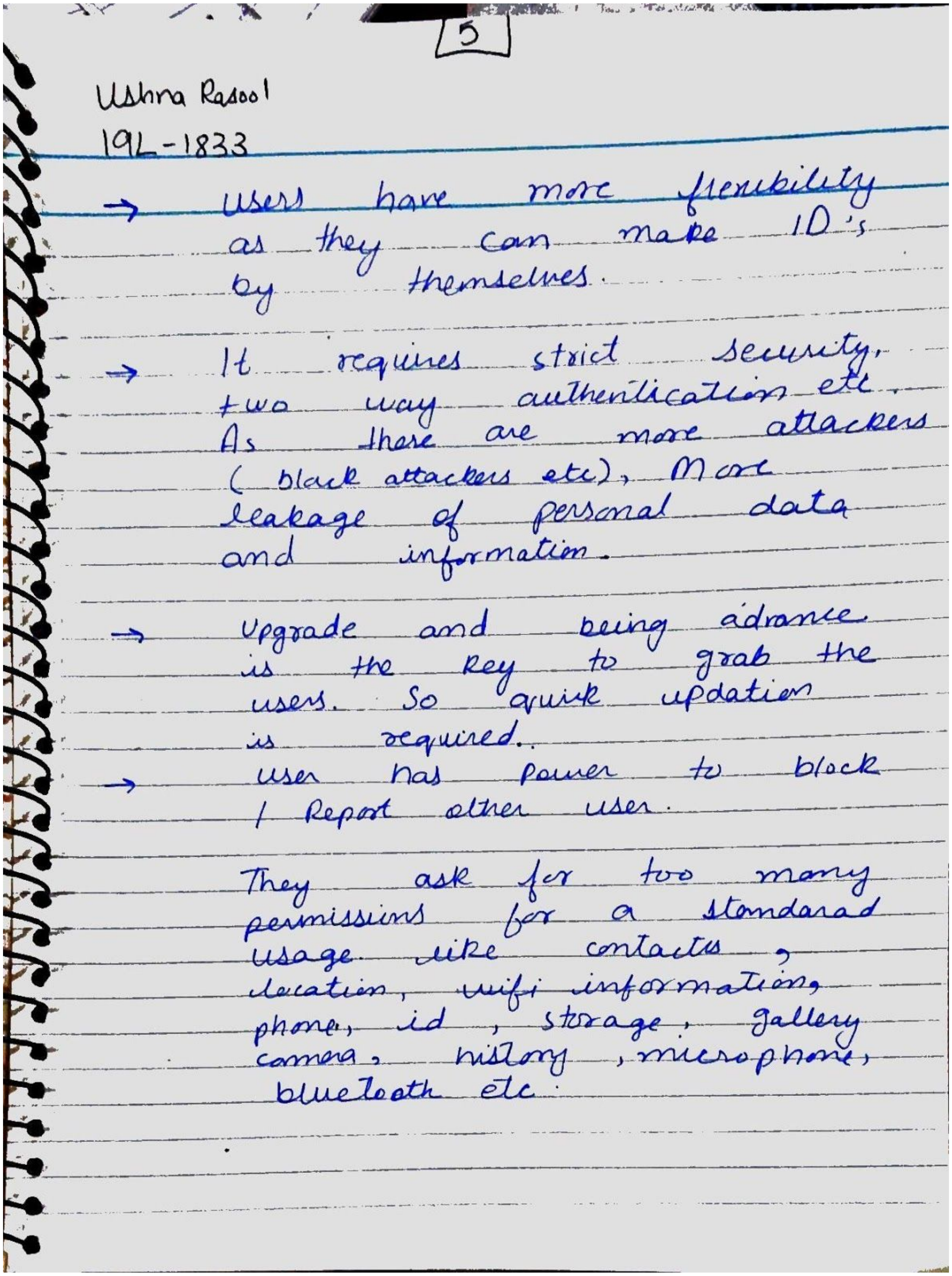
→ Need compatible memory to Store the data.

☆ Long term:-

- <3 years span
- +100 users
- Apps like Facebook, instagram. etc.

→ It requires long term planning. (like permanent Solutions, long term target etc)

→ Need more space. As facebook downloading takes 50 MB but it stores alot of Cache. If a person use Face book scrooling for one hour daily, he will use 3 GB approximately in one month.



5

Ushna Rasool
19L-1833

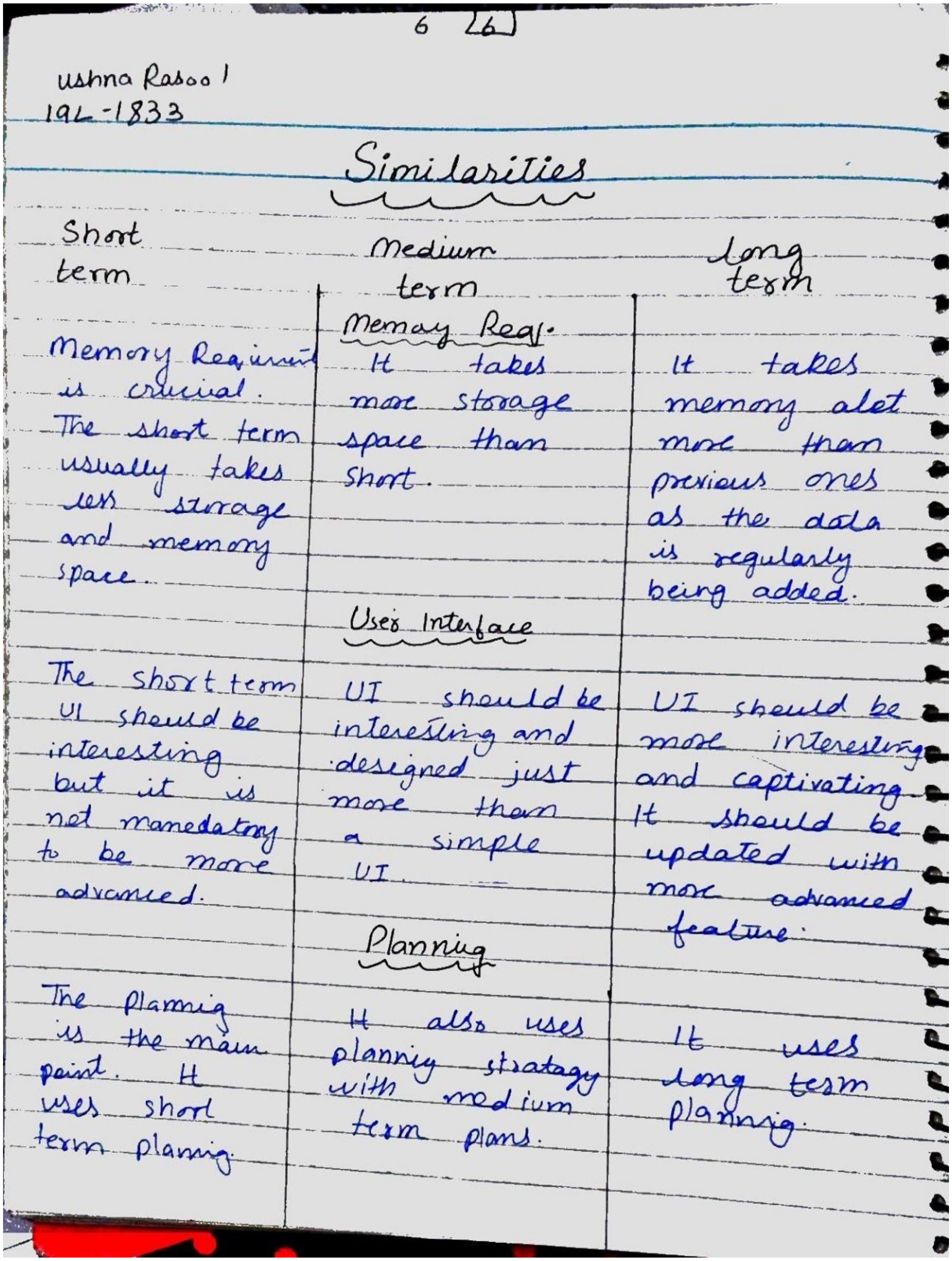
→ users have more flexibility as they can make ID's by themselves.

→ It requires strict security, two way authentication etc. As there are more attackers (black attackers etc), More leakage of personal data and information.

→ Upgrade and being advance is the key to grab the users. So quick updation is required.

→ user has power to block / Report other user.

They ask for too many permissions for a standarad usage like contacts, location, wifi information, phone, id, storage, gallery camera, history, microphone, bluetooth etc.

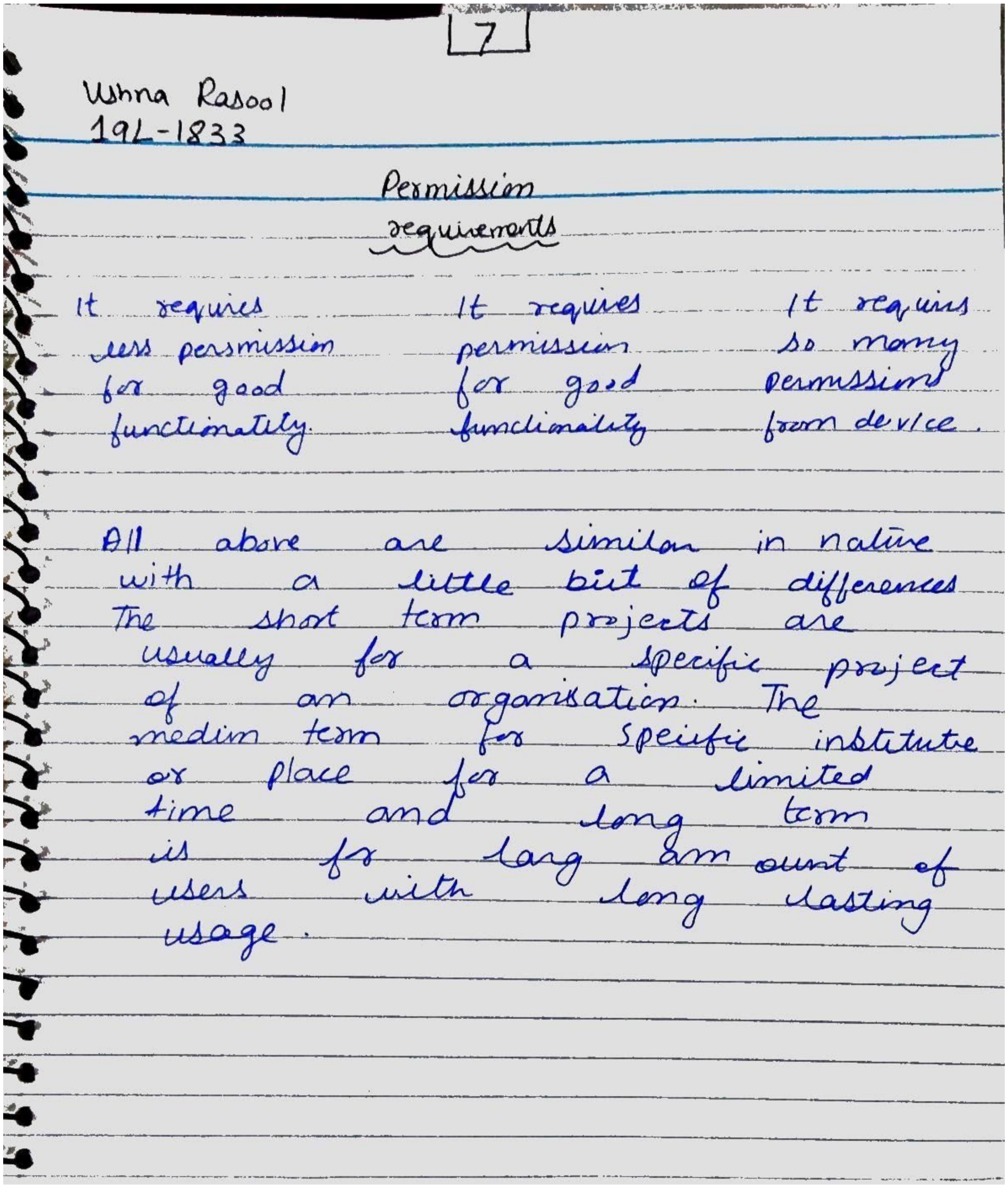


6 [6]

ushna Rasool
19L-1833

Similarities

Short term	Medium term	long term
	Memay Reqr.	
Memory Requirement is crucial. The short term usually takes less storage and memory space.	It takes more storage space than short.	It takes memory alot more than previous ones as the data is regularly being added.
	User Interface	
The short term UI should be interesting but it is not manedatory to be more advanced.	UI should be interesting and designed just more than a simple UI.	UI should be more interesting and captivating. It should be updated with more advanced feature.
	Planning	
The planning is the main point. It uses short term planing.	It also uses planning stratagy with medium term plans.	It uses long term planning.



7

Ushna Rasool
19L-1833

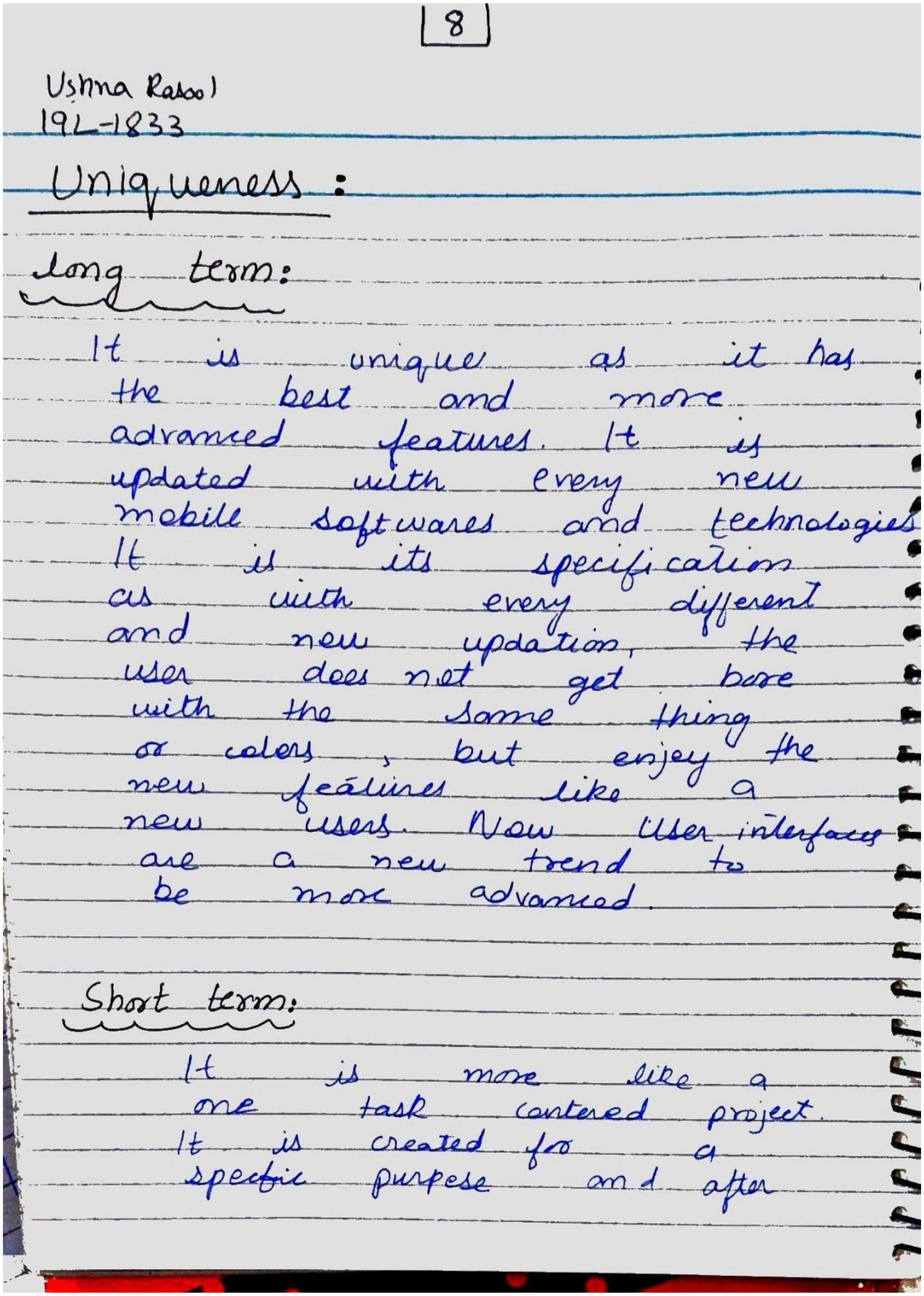
Permission requirements

It requires less permission for good functionality.

It requires permission for good funclionality

It requires so many permission from device.

All above are similar in nature with a little bit of differences The short term projects are usually for a specific project of an organisation. The medim term for speific institute or place for a limited time and long term is for larg amount of users with long lasting usage.



8

Ushna Rasool
19L-1833

Uniqueness :

long term:

It is unique as it has the best and more advanced features. It is updated with every new mobile softwares and technologies It is its specification as with every different and new updation, the user does not get bore with the same thing or colors , but enjoy the new featires like a new users. New User interfaces are a new trend to be more advanced.

Short term:

It is more like a one task centered project. It is created for a specific purpese and after

Ushna Rasool
19L-1833

the task is completed
or project is completed,
It is of no use
we can say that
it is a task oriented
software system for an
organisation to keep in
track the progress
and share information
It is a users / customers
focused.

USHNA RASOOL

19L-1833

Ushna Rasool
19L-1833

Medium term:-

The medium term projects often conducted to see the impact of their software system on the users and how much improvements are required. They are usually free in cost because of the testing phase. So, users can give the feed back and can easily users reaction reviews. The long te It is unique because these feedback and reviews are the key point to decide whether to go on with this software project or end them. If succefull feedback is recieved, the app can be converted into the long term Project.

what you have discussed was not the point of the question - you were supposed to identify RE activities such as elicitation (and its sub activities) for three types of projects.
-15

Q 2:

Introduction:

Specifying the requirements and arranging the requirements is the most nerve wracking as it is not a child play. The ambiguity in common language statements create a buzz in specifying the requirements. Many errors and loopholes were developed because of this vagueness. In the Domains like gas and oil, imaging, Power and aerospace, a company named General Designs (GI), develops, designs ,validates and deploys the infrastructure of the software. As the complexity of these intensive software systems grew, General Electric Global Research joined hands with the General Electric Aviation System to introduce a new tool ASSERT ™. ASSERT™ stands for Analysis of Semantic Specifications and Efficient generation of Requirements based tests. It uses structured natural language to associate both machine and human readable language. Its main capturing point that drew users toward itself was so close to the readable English language instead of the formal and strict ones. It helps to detect the pitfalls and loopholes by interpreting the structured NL to a formalism. An automated Theorem prover can justify it. It urges to interpret the requirements into the SMT formulas, satisfiability modulo theories. SMT can solve the requirement test based cases automatically, that's the plus point. In short ASSERT ™ proves itself as an automatic requirement test based generator and a big time saver. In this paper, the author focused on solving the common problems that engineers, developers and designers came through when encapsulating the requirements with the help of ASSERT ™ .

SUMMARY:

no a good critique
+ it should have
been and written
-5

Before ASSERT ™ , it was very hard to filter o l errors in them. But with the emergence of ASSERT ™ by GE, it is now more compatible t any projects have been run through this and shows a strong grip of handling the industrial size problems. All of these projects had functional requirements mostly but when it was user requirements , 38,272 test procedures and 19,276 use cases in an avionics project, it shows no errors with 1002 requirements being auto generated. It shows more growth in finding requirements on an avionics compute platform. At first, due to its error checking feature, it shows a climax in producing requirements that are more mature than a plain text requirement. Secondly, it suppressed the design part because with the requirement findings the functional architecture is already being captured. It also produce automated test cases that verified the requirements as soon as they were fetched. The test cases and techniques are discussed here completely. It shows that the experts of the domain while using ASSERT ™ don't need to apply analysis or formal modeling to get the advantage from the formal techniques and automatic test generation of their requirements.

CRITIQUE:

The requirements engineer confronts several problems and difficulties while finding and pen down the requirements. Ideally there is much more than writing a requirement. It takes deep insight of what is important, what to conclude, how to refine these requirements further. Normally people think that their native language is the easiest one as they have been speaking to them since forever. But the problem is that the native natural language is somehow ambiguous and vague. It's hard to filter out the desired requirements. Let's take a look at the author point of view of positive things that ASSERT ™ can do.

To shrink the desired requirements, a complete balance must be sought out between natural language. The requirements should be represented in terms of domain and domain ontology helps to derive a domain specific language for the RE expression. A formal representation is formed by the combination of ontology language and requirements language making a subset of set theory and first order logic. So these requirements are now applicable to the methods like analysis and for automatic generation of the test cases. As the main cause of ASSERT ™ is to make the output as useful as possible for the user, it used different approaches like error marking, informational marker, warning marker etc. these are convenient for the writer to be conscious of the ambiguities and errors that are written and also those assumptions that have been put up to avoid the vagueness.

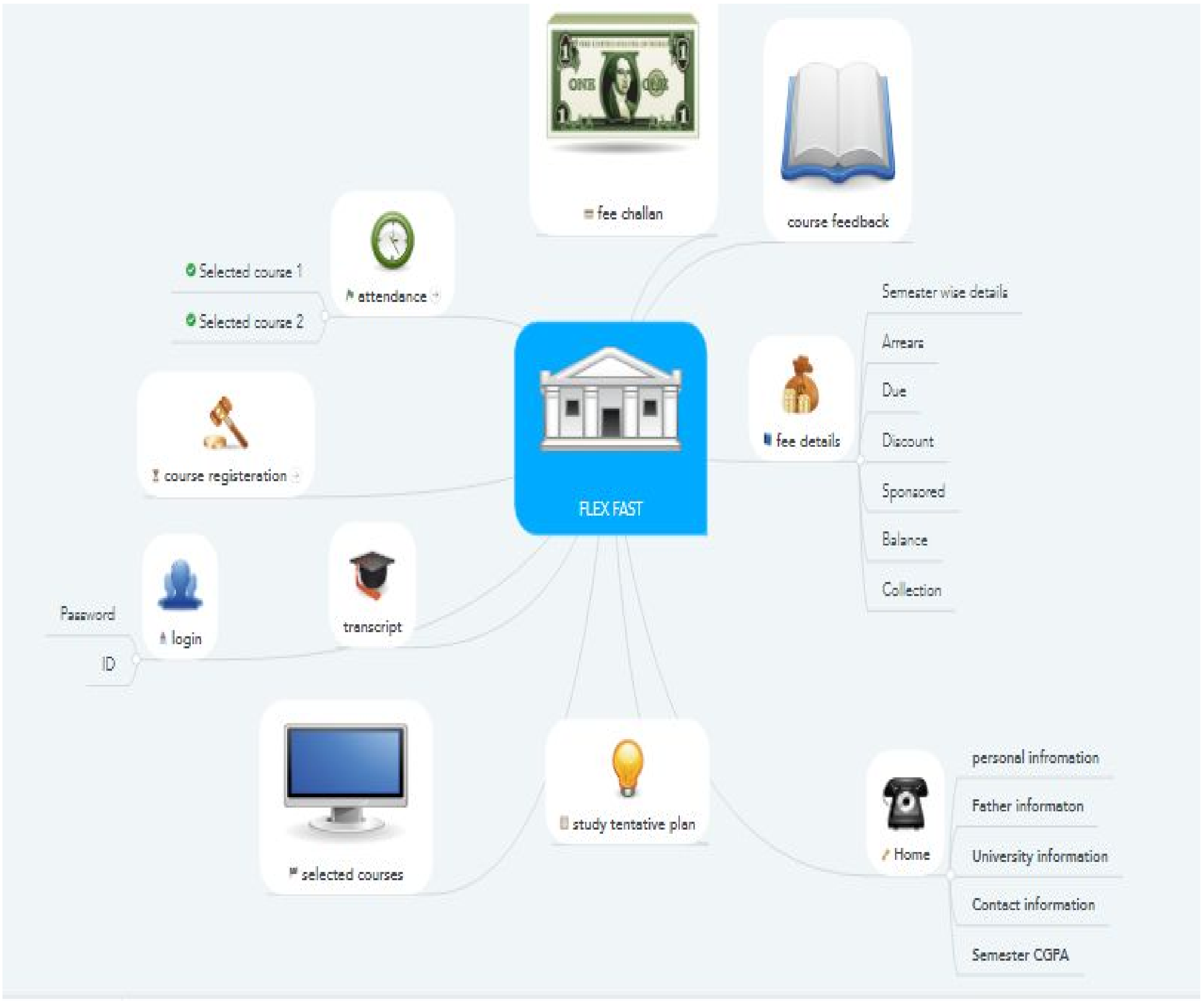
While writing requirements, they are tumbled down from high level requirements to low level requirements.a requirement telling device should indicate the extra details of the lower level requirements rather than focusing on high level

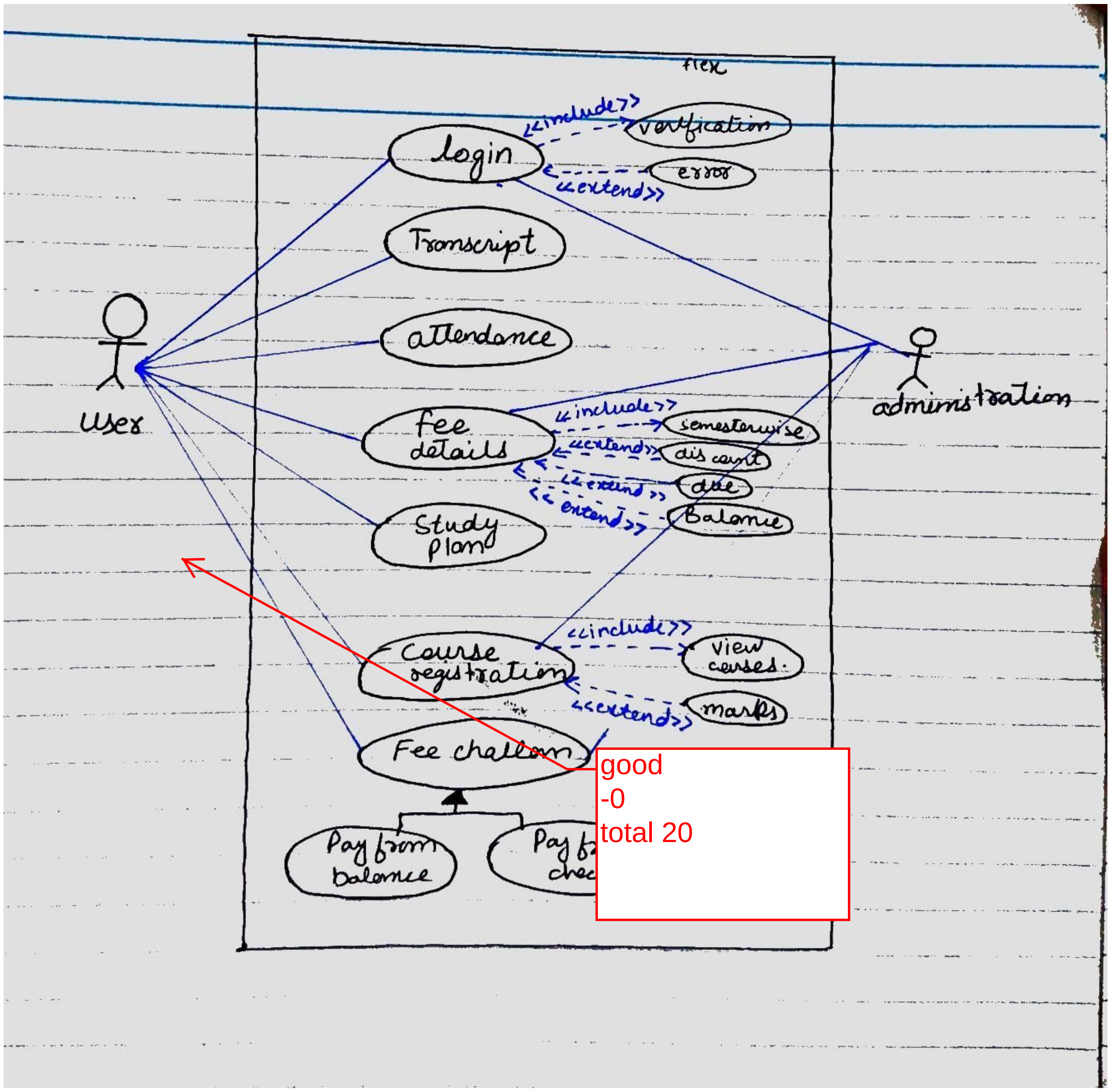
requirements. This is done by the process called decomposition by th ASSERT ™ . This process indicates that the specification and descriptive documentation of the low level design and requirements.
Another loophole for penning down the requirements was what an under production/design system should do. controlling all the requirements in a consistent ontology manner is called type checking. Type checking errors occur when the things happen when they are not in the lists thus making sure that the list should go in a basic consistent manner. But type checking sometimes can be very challenging. We often use short phrases in native language that are ambiguous. So ASSERT ™ uses implied properties to give sense to those short phrases. Like we say he is 30 it can have several meanings like if his age is 30 or his weight is 30? So ASSERT ™ helps to interpret the meaning and makes sense.
For big projects, the requirements are trumple down into several subsystems thus causing developers a problem to know their dependencies. ASSERT ™ solves the problem by managing these assumptions like it can assume the values of the property based on other subsystems.
In this paper, the author nicely jotted down the example of an aircraft engine sensor. It uses dual channel pressure sensor Px and selected values are evaluated by the received inputs shown in a figure 1 of the paper. A sparse skeleton is described nicely with its properties and restrictions. ASSERT™ used to read the models.
To perform the formal analysis, requirements are required and captured by SADL Requirements Language. Our target is to squeeze the distance between the target language and subject matter experts' mental model, to convert natural language to formal language. ASSERT™ is eligible to take the domain as a semantic model and this model is based on SADL. SADL is a controlled English language and a rich IDE to test, create and maintain the semantic model. The requirements are analysed by Requirement Analysis Engine (RAE) of ASSERT™. RAE localise the error, make a report of the error if found with proof and produce test cases.
After requirement analysis by RAE, the requirements are processed by the Automated Test Generation tool of ASSERT™ (ATG) that tests the requirement with the test coverage of pre-defined regulation and produces test cases and techniques without a single line of code.
The ASSERT ™ shows enormous growth but still many more errors are yet to be solved. It is expected from ASSERT ™ to solve all the problems and shrink the mental efforts of the requirements writer.

Conclusion

In this paper the prior loopholes were discussed and then solved by ASSERT™. Before, there was a huge gap between the introduction of the error and discovery of the error that caused the increase in the cost as it also required the architectural changes and redesigning of the system. But ASSERT™ solves all the problems by producing the conflict free requirement, formal analysis and resolving of errors thus giving the developer a huge relief as they are needed to do analysis and modeling by themselves.

Q3:


FLEX FAST
fee challan
course feedback
attendance
Selected course 1
Selected course 2
course registeration
login
Password
ID
transcript
selected courses
study tentative plan
fee details
Semester wise details
Arrears
Due
Discount
Sponsored
Balance
Collection
Home
personal infromation
Father informaton
University information
Contact information
Semester CGPA



good
-0
total 20