

4. Non-functional requirements

4.1 Performance Requirements:

A regular check on the system will take place to ensure that the software is properly functional and also to detect if the system crashed, hanged or an operating system error occurred.

4.2 Safety Requirements:

The user databases will be backed up regularly to ensure no loss of data.

4.3 Security Requirements:

The software can be used only by authorized people i.e people with the correct combination of username and password.

4.4 Software Quality Attributes:

Reliability:

The software will be tested and debugged multiple times in different scenarios before finally releasing it. Thus giving the user assurance and reliability on the software.

Correctness and Accuracy:

The software will give results only based on the classification algorithms and will not yield results that are biased.

Testability:

The software would be easy to test and find defects.

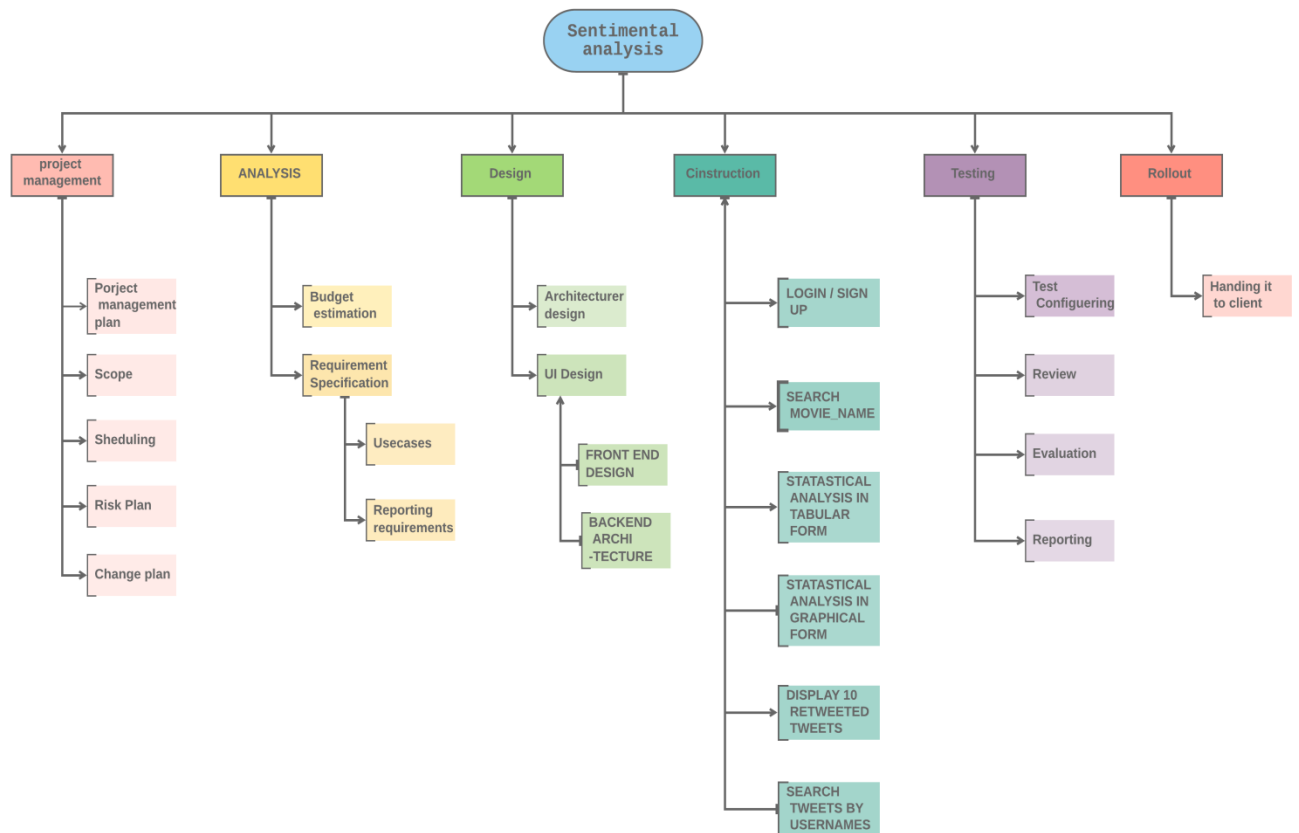
Security:

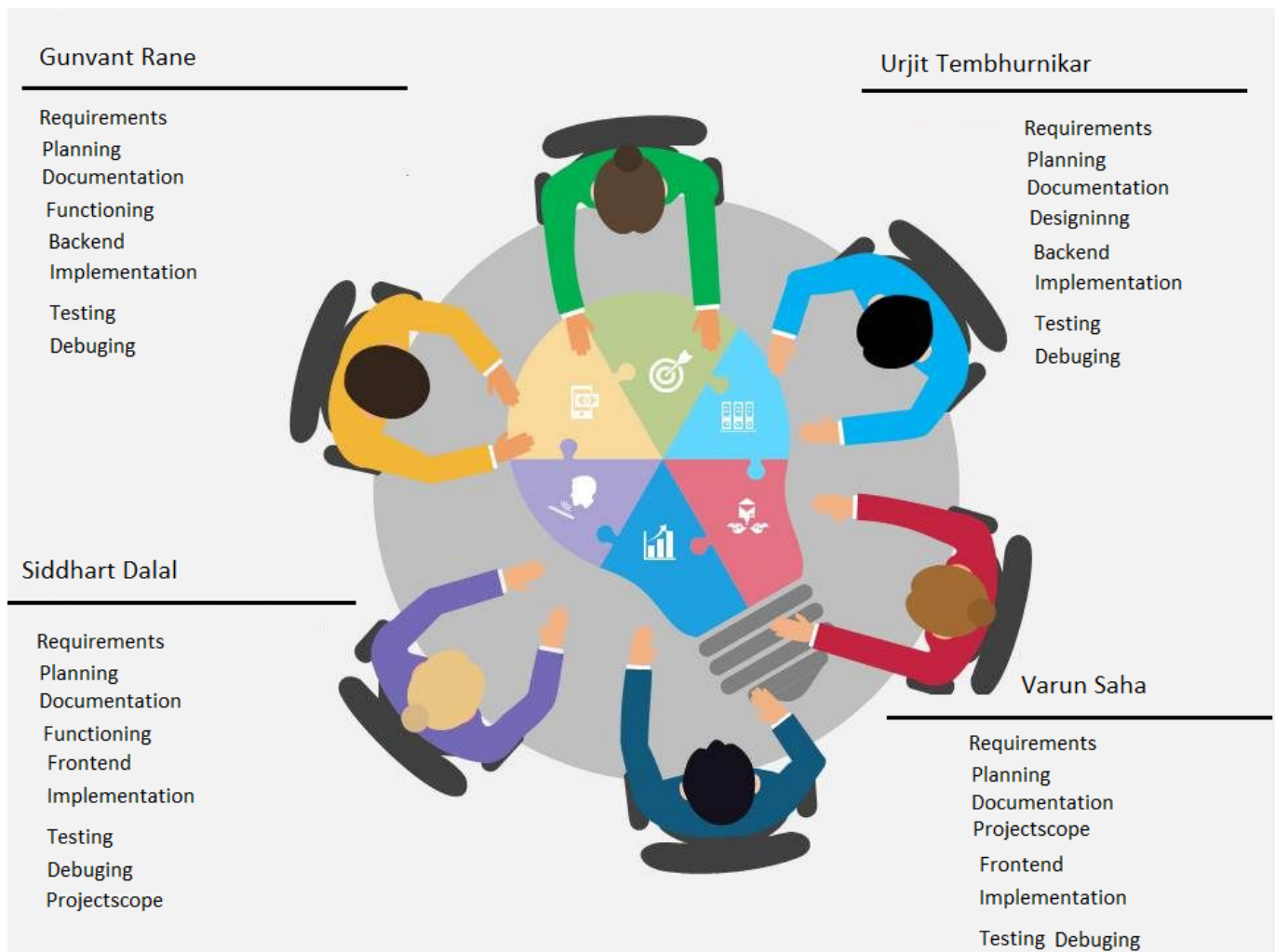
Only the right combination of username and password would allow one to access the software.

5. Project scheduling and estimation

5.1 Functional decomposition (WBS)

A work-breakdown structure (WBS), also referred to as "Contract Work-Breakdown Structure" or "CWBS", in project management and systems engineering, is a deliverable-oriented breakdown of a project into smaller components. A work breakdown structure is a key project deliverable that organizes the team's work into manageable sections.





The team members are as follows:

- A: Gunvant Rane
- B: Urjit Tembhornikar
- C: Siddharth Dalal
- D: Varun Saha

S.no	Tasks	Roles and Responsibilities
1.	Requirements gathering.	Gunvant,Urjit,Siddhart,Varun
2.	Planning and Software requirements.	Gunvant,Urjit,Siddhart,Varun
3.	Project scope	Siddhart ,Varun
4.	Documentation.	Gunvant,Urjit,Siddhart,Varun
5.	Designing.	Urjit , Varun
6.	Developing functional specifications.	Gunvant,Siddhart
7.	Frontend coding	Siddhart ,Varun
8.	Backend coding	Gunvant,Urjit
9.	Implementation	Gunvant,Urjit,Siddhart,Varun
10.	Testing and Debugging	Gunvant,Urjit,Siddhart,Varun

Task Duration :

S.no	Task	Timespan
I.	Phase I	1 Month
	1.Introduction	1 week
	2.Overall description	1week
	3.Functional requirements	2 week
II.	Phase II	1 Month
	1.Project scheduling and estimation	2 week
	2.Non-functional requirements	2 week
III.	Phase III	2 week
	1.Analysis models	1 week
	2.Implementation and screenshots	1 week
IV.	Phase IV	2 week
	1.Test cases	1 week
	2.Conclusion and future-scope	1 week

5.3 Task Duration (GANTT Chart)

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities .

- A Gantt chart helps us examine:
- What the various activities are
- When each activity begins and ends
- How long each activity is scheduled to last
- Where activities overlap with other activities, and by how much
- The start and end date of the whole project

Fig 01:

The overview of the task is shown in figure.

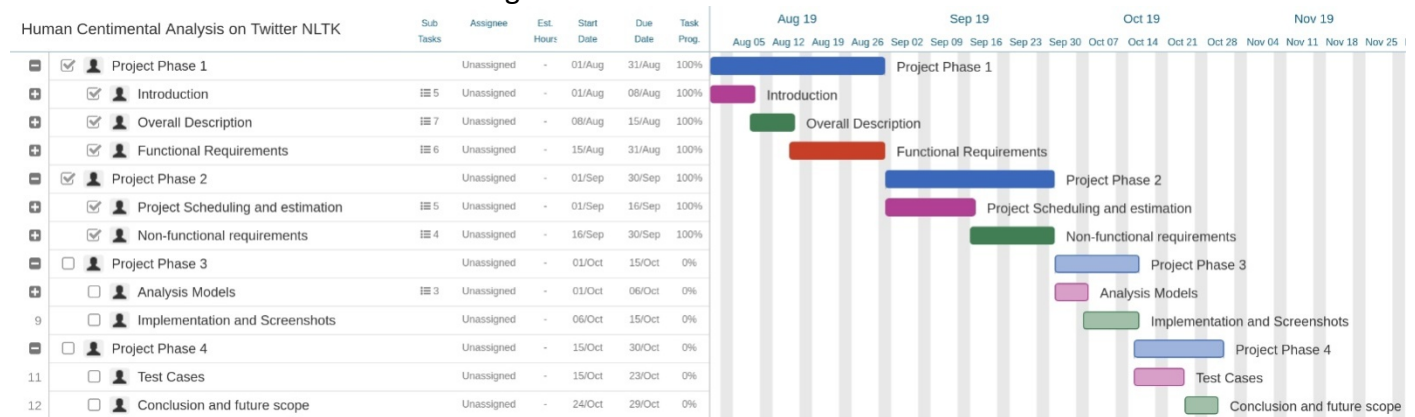
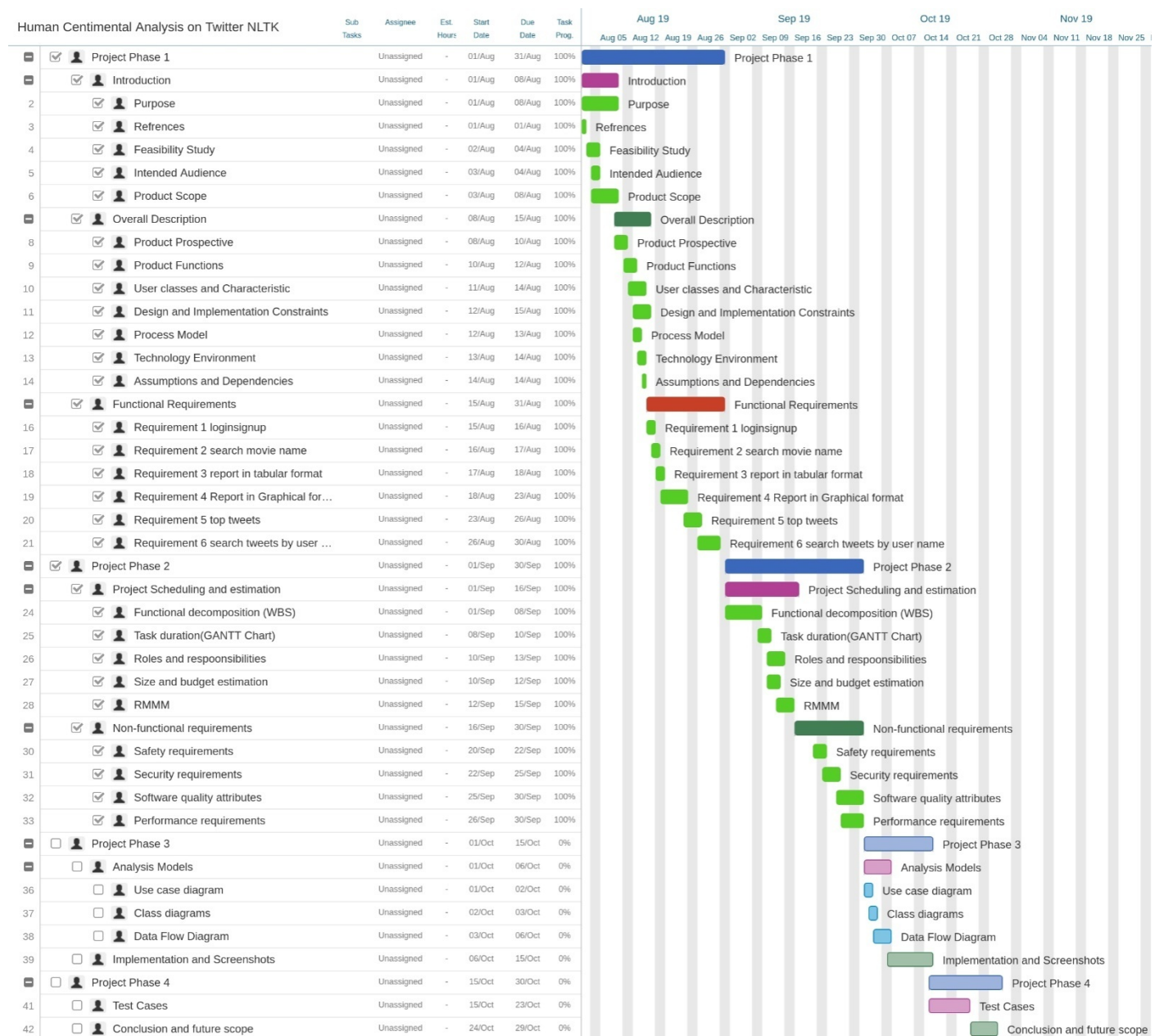


Fig 02 :

The detailed breakdown of the project is shown in figure. The task and subtask are clearly shown.



5.4 Size and Budget Estimation

Our team has divided the project into six main functionalities along with their estimates of size of functionality shown below:

Effort Estimation :

$Ev = (\text{optimistic} + 4 * \text{Most Likely} + \text{Pessimistic}) / 6$

Cost Estimation :

Assume cost to be Rs 15 per LOC

Sr. no	Name of function	Optimistic	Most likely	Pessimistic
1.	Login	280	297	310
2.	Signup	650	664	670
3.	Search	125	130	138
4.	Twitter	515	538	555
5.	Get sentiments	236	250	271
6.	Tabular form	46	50	59
7.	Graphical from	70	64	87

Effort :

1. $(280+297+310)/6 = 147.83$ LOC

2. $(650+664+670)/6 = 330.66$ LOC

3. $(125+130+138)/6 = 65.5$ LOC

4. $(515+538+555)/6 = 268$ LOC

5. $(236+250+271)/6 = 84.5$ LOC

6. $(46+50+59)/6 = 25.83$ LOC

7. $(70+64+87)/6 = 36.83$ LOC

Total = $147.83 + 330.66 + 65.5 + 268 + 84.5 + 25.83 + 36.83$

= 959.15 LOC

= **960 LOC roundup**

Cost :

Assume cost to be Rs 15 per LOC

$15 \times 960 = \underline{\underline{14,400 \text{ inr}}}$

5.5 RiskMonitoring,MitigationandManagement(RMMM)

Problems encountered:

Customer feedback detection:

Mitigation: Customer dissatisfaction over the feedback system.

Monitoring: Interaction with the selected customer base and deducing a conclusion as to what fits best for all.

Management: Using better technology which is feasible as well as acceptable by the customer base.

Performance issue:

Mitigation: Using a suitable application for designing and developing the software.

Monitoring: Careful analysis of each data input by the customer.

Management: Interacting with the customer user base and addressing their grievances.

Project task extending beyond deadline:

Mitigation: Estimating the cost in late delivery of the particular product. Steps need to be taken to ensure timely submission of the product before the deadline.

Monitoring: Keeping a careful watch on the team's recent developments and constructing a timeline for the completion of the. Project.

Management: Organizing a meeting between the developer and the customer to ensure the product meets the customer requirements.

Memory allocation problems:

Mitigation: The memory storage may cause some problems at a later stage.

Monitoring: Keeping a careful watch on the progress made and servicing all the issues.

Management: Organizing meetings (both formal and informal) with all the group members and addressing the issue one after the other.

Risk Summary	Risk Category	Probability	Impact (1-4)	RMMM
Customer Feedback Detection	Technical	0.25	2	<p>Mitigation: Customer dissatisfaction over the feedback system.</p> <p>Monitoring: Interaction with the selected customer base and deducing a conclusion as to what fits best for all.</p> <p>Management: Using better technology which is feasible as well as acceptable by the customer base.</p>
Performance issue:	Technical	0.48	3	<p>Mitigation: Using a suitable application for designing and developing the software.</p> <p>Monitoring: Careful analysis of each data input by the customer.</p> <p>Management: Interacting with the customer user base and addressing their grievances.</p>
Project task extending beyond deadline:	Known	0.19	2	<p>Mitigation: Estimating the cost in late delivery of the particular product. Steps need to be taken to ensure timely submission of the product before the deadline.</p> <p>Monitoring: Keeping a careful watch on the team's recent developments and constructing a timeline for the completion of the. Project.</p> <p>Management: Organizing a meeting between the developer and the customer to ensure the product meets the customer requirements.</p>
Storage allocation problems:	Technical	0.42	3	<p>Mitigation: The memory storage may cause some problems at a later stage.</p> <p>Monitoring: Keeping a careful watch on the progress made and servicing all the issues.</p> <p>Management: Organizing meetings (both formal and informal) with all the group members and addressing the issue one after the other.</p>
Sale of a particular product:	Unpredictable	0.44	4	<p>Mitigation: The technical team and the sales team should be constant contact with one another. The sales team should be aware of all the features being provided and target those properties which are marketable.</p> <p>Monitoring: If both teams are in constant contact then, step by step deduction of all the facilities being provided can be utilized.</p> <p>Management: By holding meetings, various discussions about the progress can be held and improved clarity can be brought upon the final product to be delivered.</p>