

“Novel For Requirement Management and Traceability Module”

AMRUT JAMDAR¹, AKSHAY MAHADIK², TEJAS NIMASE³, MAHESH DHIMDHIME⁴

¹ Student, Computer, JSPM's JSCOE, Maharashtra, India.

² Student, Computer, JSPM's JSCOE, Maharashtra, India.

³ Student, Computer, JSPM's JSCOE, Maharashtra, India.

⁴ Student, Computer, JSPM's JSCOE, Maharashtra, India.

Abstract

Project requirements writing is a very important role in every Software organization, yet relatively little research effort has focused on how to write them. This Project takes note on issues arising in requirements and their traces to other requirements and presents an approach to requirement writing, traceability, team management, and describes agenda to facilitate the use of our project approach. Requirement writing plays very important role in successful and smooth flow project completion. A framework for writing requirements is introduced. A complete tool set, based on this framework, has been prototyped. The components of this system are a requirement suite tool that gathers project requirement information, a grouping tool that groups various requirements, a team tool that stores team members working for a project, and a traceability tool that traces different requirements

Keywords: Traceability, Data Mining, File Processing/File System, Java/JavaFx, Artificial Intelligence, Genetic Programming, MySQL.

I. INTRODUCTION

The current world is becoming more dependent on software intensive systems. It is the essential capability that can bridge the two perspectives. The RE research field builds the effective bridges between these and other sub disciplines of the CS and the IT. RE helps stakeholders communicate, helping to reconcile their conflicts, clarify their goals, and reflect their priorities. It is process of documenting, analysing tracing, prioritizing and agreeing on requirement and then controlling change. In our Project a framework for writing requirements is introduced.

II. . PROBLEM STATEMENT

Many software organizations lack the purpose of writing requirements into a systematic way. Either the management or the coordinator lacks the way of requirements writing. Even tough if he/she writes requirements, not all proper Traceability is done. Various organizations use softwares such as Word, Excel but they don't do it systematically. Sometimes, they do requirements writing properly but they lack the traceability, which results in messing up the art of requirements writing. Not only that, at the end, they do manual and laborious work of analysing the requirements to get the expected results in terms of how many requirements are written, how may traces are there, how many requirements can be dependent and independent. Our project of Requirement Suite Tool with various features helps this problematic scenario.

III. PROPOSED SYSTEM

(a) System Architecture

Proposed system is responsible for managing requirements of any project which are important or responsible for development of that project. Initially the system is provided with the traceability and based on that we can trace the

requirements present that project as well as we can able to trace any requirement with other requirement or group of requirements except same one . The proposed system makes use of one of the best algorithms which is **Hash Table Store Algorithm, Sorting Algorithm and Traceability Algorithm** which makes efficient use of our software.

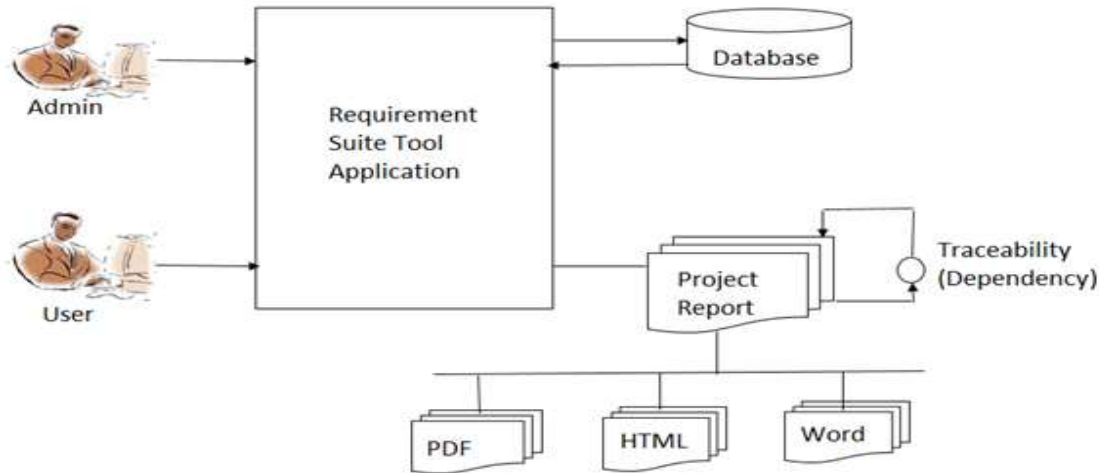
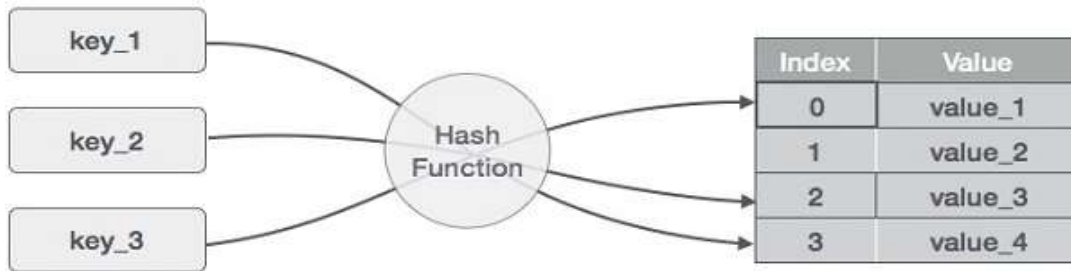


Fig. System Architecture – Requirement Suite Tool Application

(b) Algorithm Description

b.1) Hash Table Store Algorithm:

Hashing is a technique to convert a range of key values into a range of indexes of an array. We're going to use modulo operator to get a range of key values. Consider an example of hash table of size 20, and the following items are to be stored. Item are in the (key,value) format.

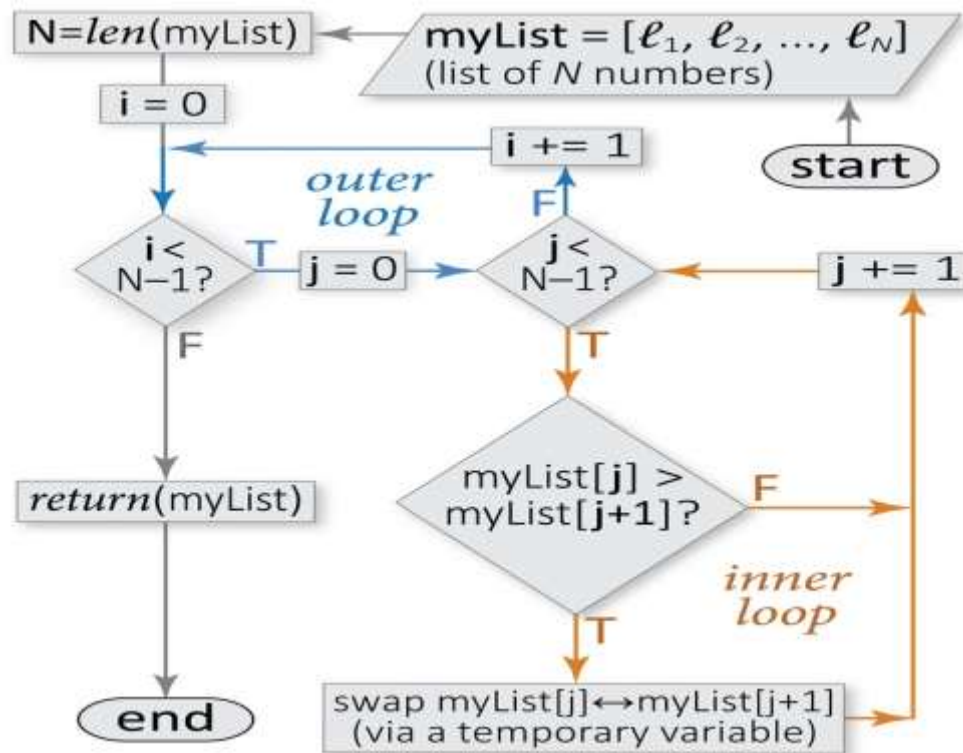


b.2) Sorting Algorithm:

In [computer science](#), a **sorting algorithm** is an [algorithm](#) that puts elements of a [list](#) in a certain [order](#). The most frequently used orders are [numerical order](#) and [lexicographical order](#). Efficient [sorting](#) is important for optimizing the efficiency of other algorithms (such as [search](#) and [merge](#) algorithms) which require input data to be in sorted lists. Sorting is also often useful for [canonicalizing](#) data and for producing human-readable output. More formally, the output of any sorting algorithm must satisfy two conditions:

1. The output is in non decreasing order (each element is no smaller than the previous element according to the desired [total order](#));
2. The output is a [permutation](#) (a reordering, yet retaining all of the original elements) of the input.

Further, the input data is often stored in an [array](#), which allows [random access](#), rather than a stack, which only allows [sequential access](#); though many algorithms can be applied to either type of data after suitable modification.

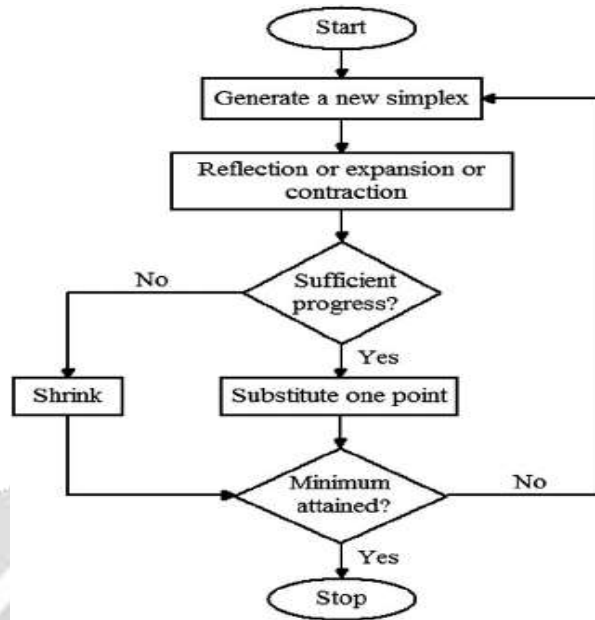


b.3) Traceability Algorithm:

Traceability is the capability to trace something. In some cases, it is interpreted as the ability to verify the history, location, or application of an item by means of documented recorded identification.

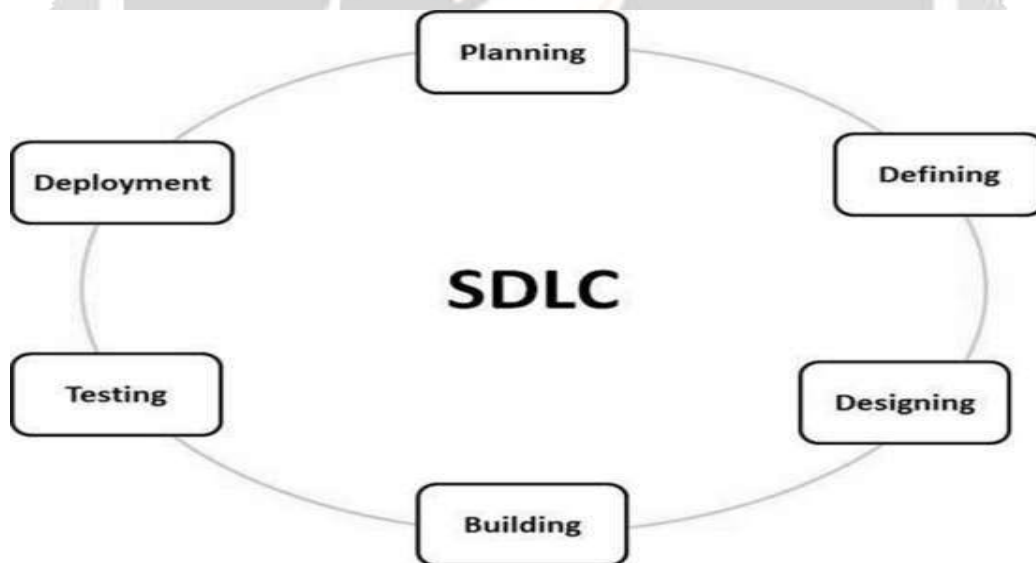
Other common definitions include the capability (and implementation) of keeping track of a given set or type of information to a given degree, or the ability to chronologically interrelate uniquely identifiable entities in a way that is verifiable.

Traceability is applicable to measurement, supply chain, software development, healthcare and security.



(C) Model Used:

c.1) SDLC:



1. **Planning:** Without the perfect plan, calculating the strengths and weaknesses of the project, development of software is meaningless. Planning kicks off a project flawlessly and affects its progress positively.

2. **Analysis:** This step is about analyzing the performance of the software at various stages and making notes on additional requirements. Analysis is very important to proceed further to the next step.

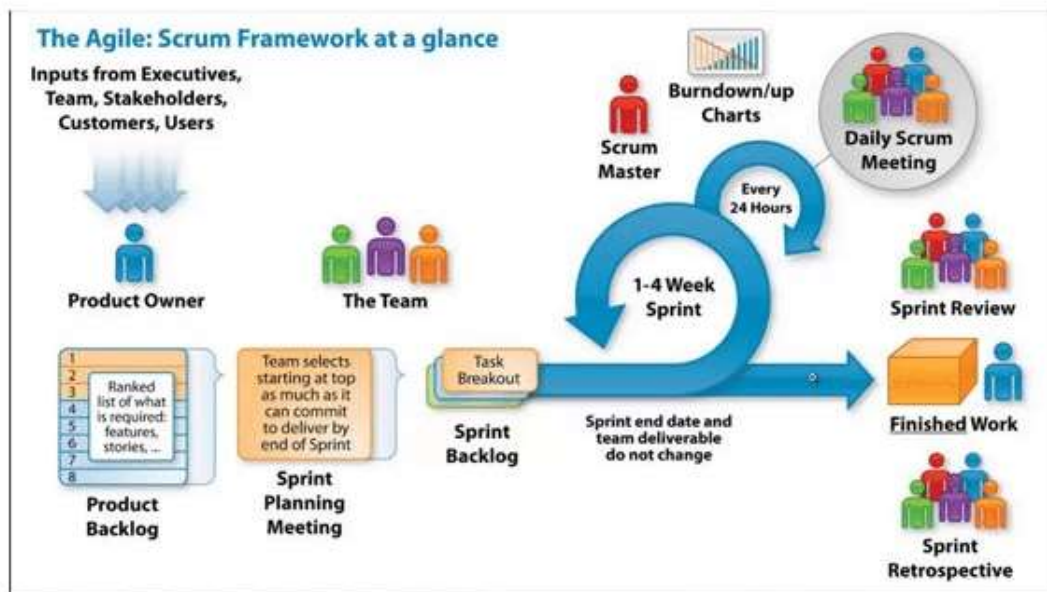
3. **Design:** Once the analysis is complete, the step of designing takes over, which is basically building the architecture of the project. This step helps remove possible flaws by setting a standard and attempting to stick to it.

4. **Development & Implementation:** The actual task of developing the software starts here with data recording going on in the background. Once the software is developed, the stage of implementation comes in where the product goes through a pilot study to see if it's functioning properly.

5. **Testing:** The testing stage assesses the software for errors and documents bugs if there are any.

6. **Maintenance:** Once the software passes through all the stages without any issues, it is to undergo a maintenance process wherein it will be maintained and upgraded from time to time to adapt to changes. Almost every software development Indian company follows all the six steps, leading to the reputation that the country enjoys in the software market today.

c.2) AGILE SCRUM:



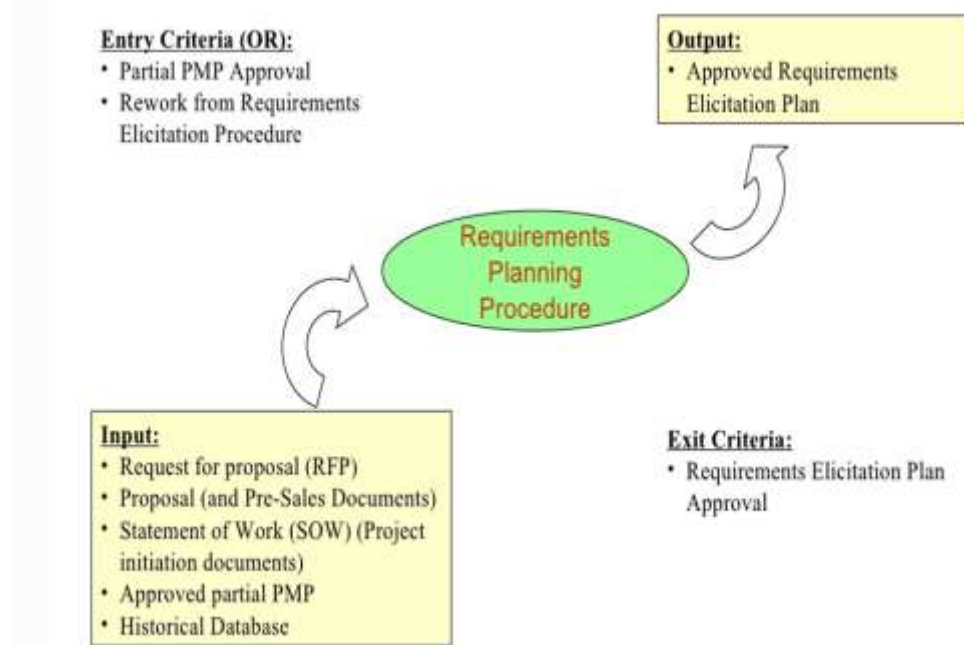
Agile Framework Model is an umbrella term for several iterative and incremental software development approaches, with each of those variations being its own *Agile framework*.

Scrum is an [agile](#) framework for managing knowledge work, with an emphasis on [software development](#). It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute [stand-up meetings](#), called *daily scrums*.

Scrum which is a specialized agile method, or we can say specialized Incremental Development process uses the term Sprint for its iterations, that is one development cycle in Scrum is called a **Sprint**. Sprint is Scrum specific, hence Sprint is an iteration but not all forms of Iterations are Sprints.

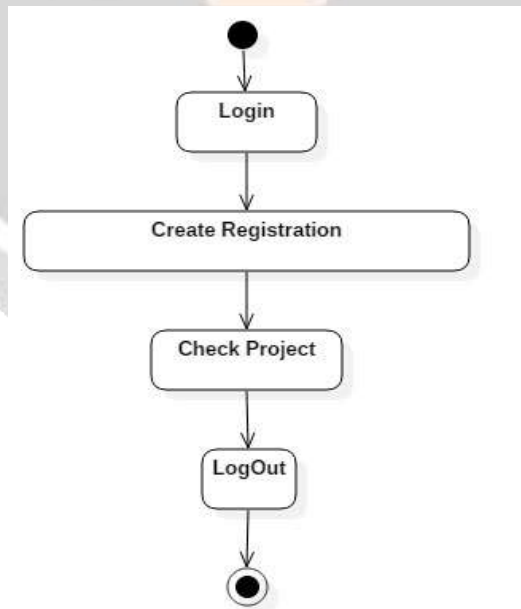
IV. ANALYSIS MODEL

(A) FLOW OF PROJECT :

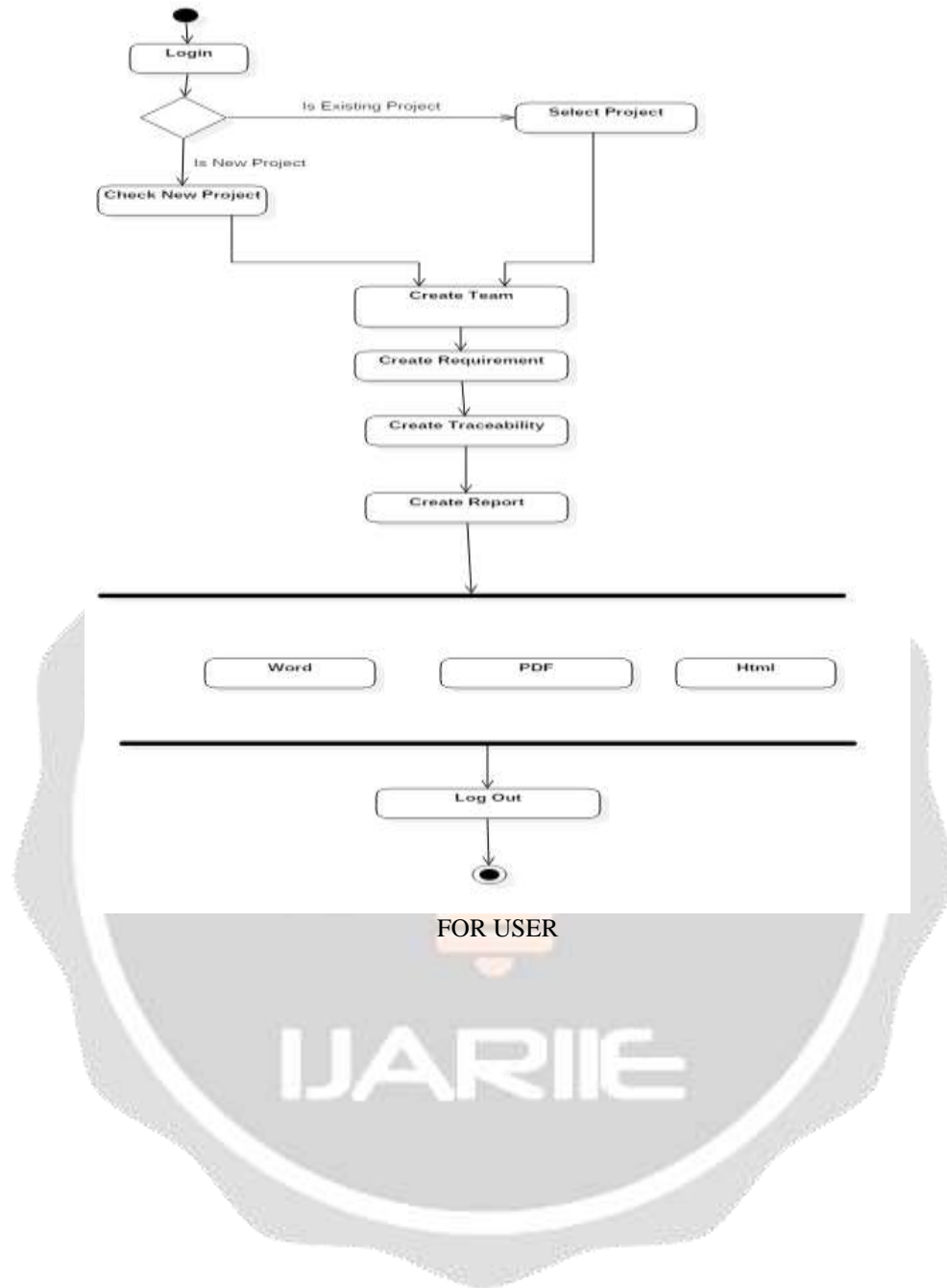


V. UML DIAGRAM

(A) ACTIVITY DIAGRAM :



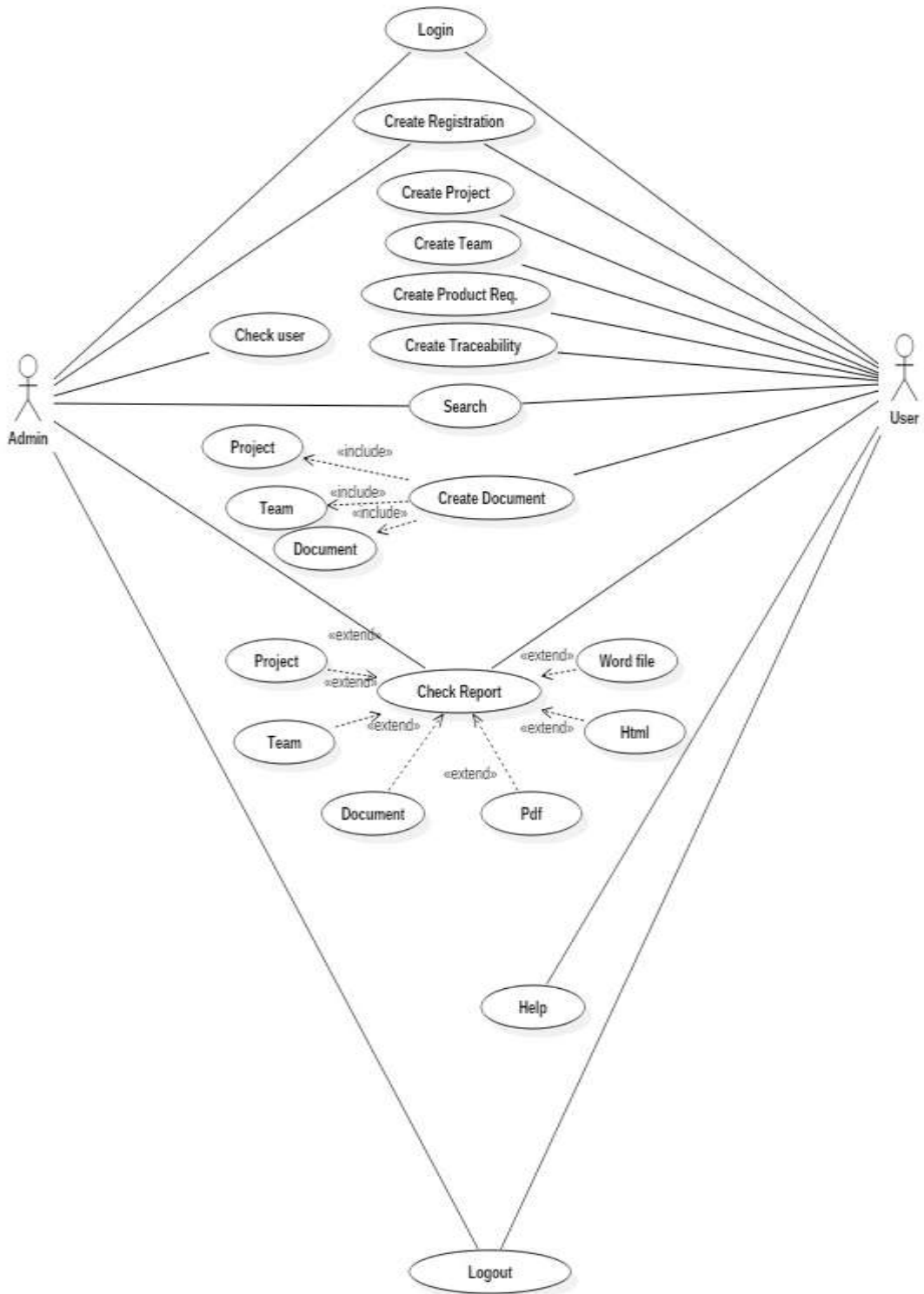
FOR ADMIN



FOR USER

IJARIE

(B) Use Case Diagram:



VI. MATHEMATICAL MODEL

Requirements are stored as R..... (1)
 $R = \{R1, R2, \dots, Rn\}$

We create groups as G.....(2)
 $G = \{G1, G2, \dots, Gk\}$

Where
 $G_i = \{R_j\}$ where i is from 1 to k from (1) and j is between 1 to n from (2)

We create traces as Tr between Requirements as.....(3)
 $Tr = \{Tr, i \rightarrow \{R1, R2, \dots, Rn\}\}$ where i is from 1 to n

We create traces as Tg between Requirements and Groups as.....(4)
 $Tr = \{Tg, i \rightarrow \{G1, G2, \dots, Gk\}\}$ where i is from 1 to k

VII. APPLICATIONS

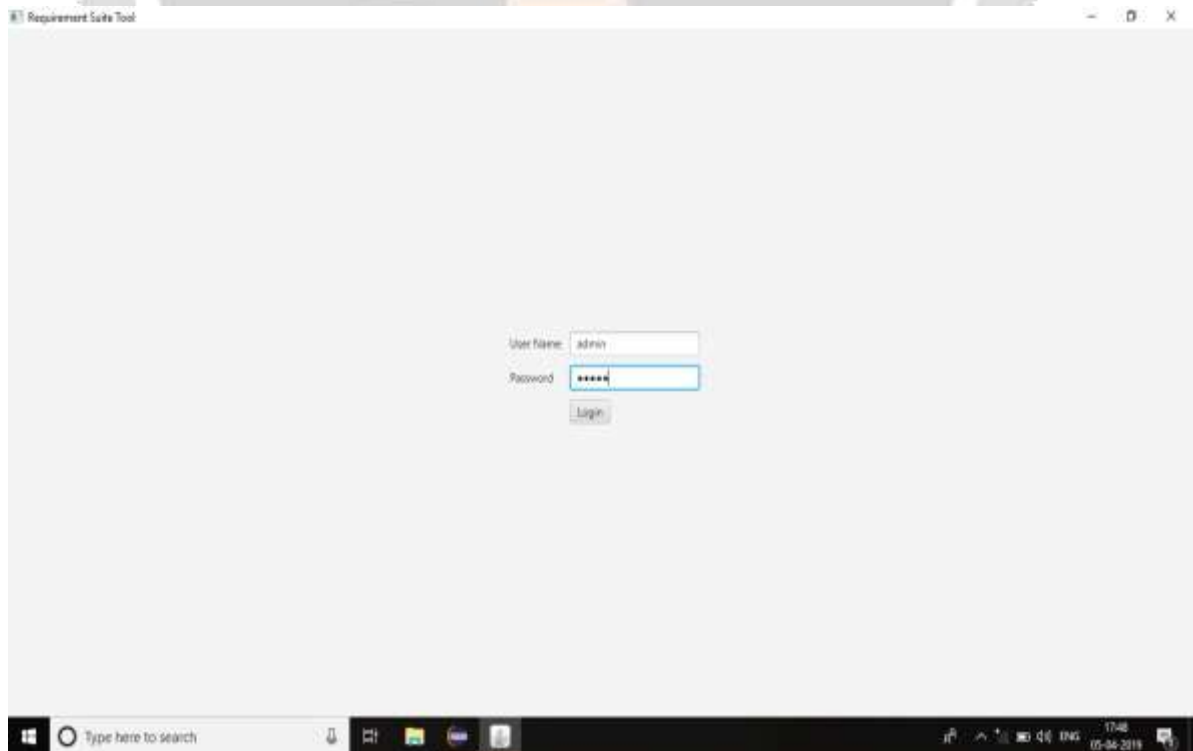
It can be used in different fields where requirement management and traceability is needed such as:

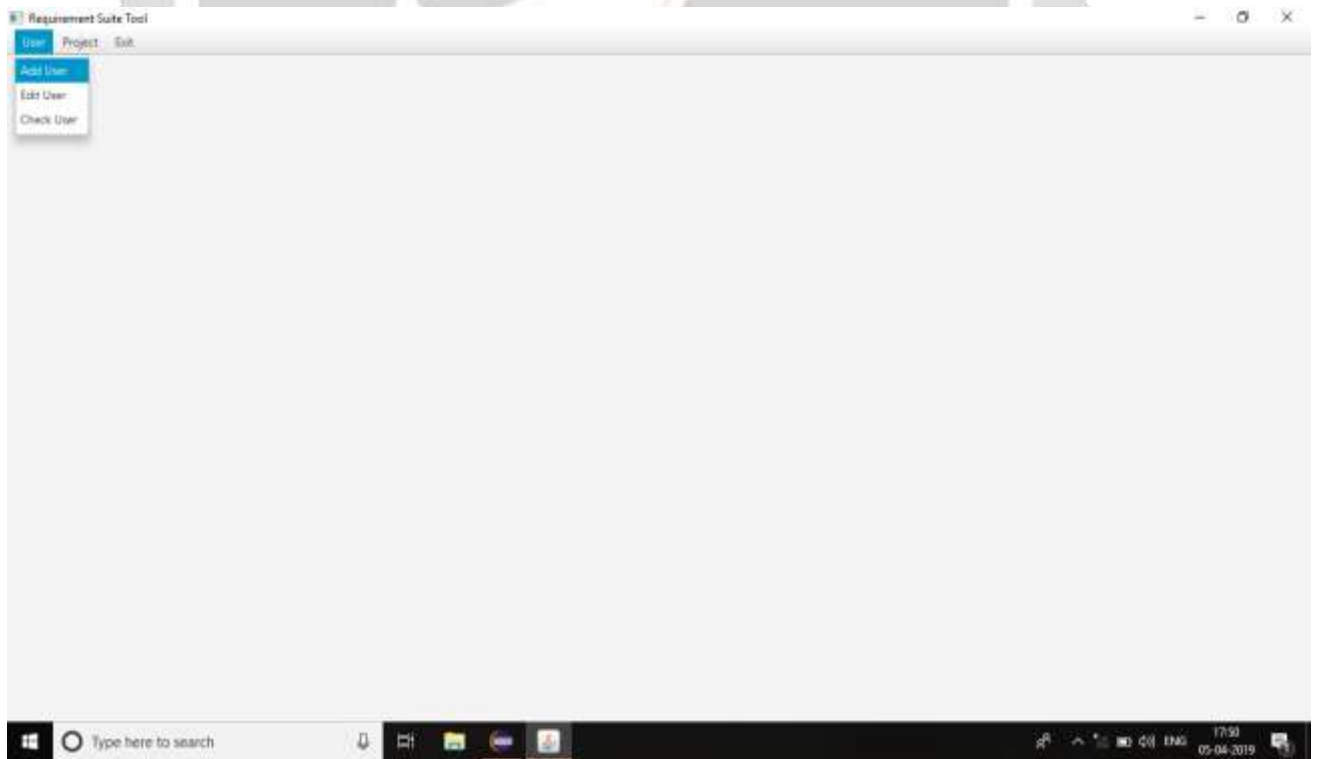
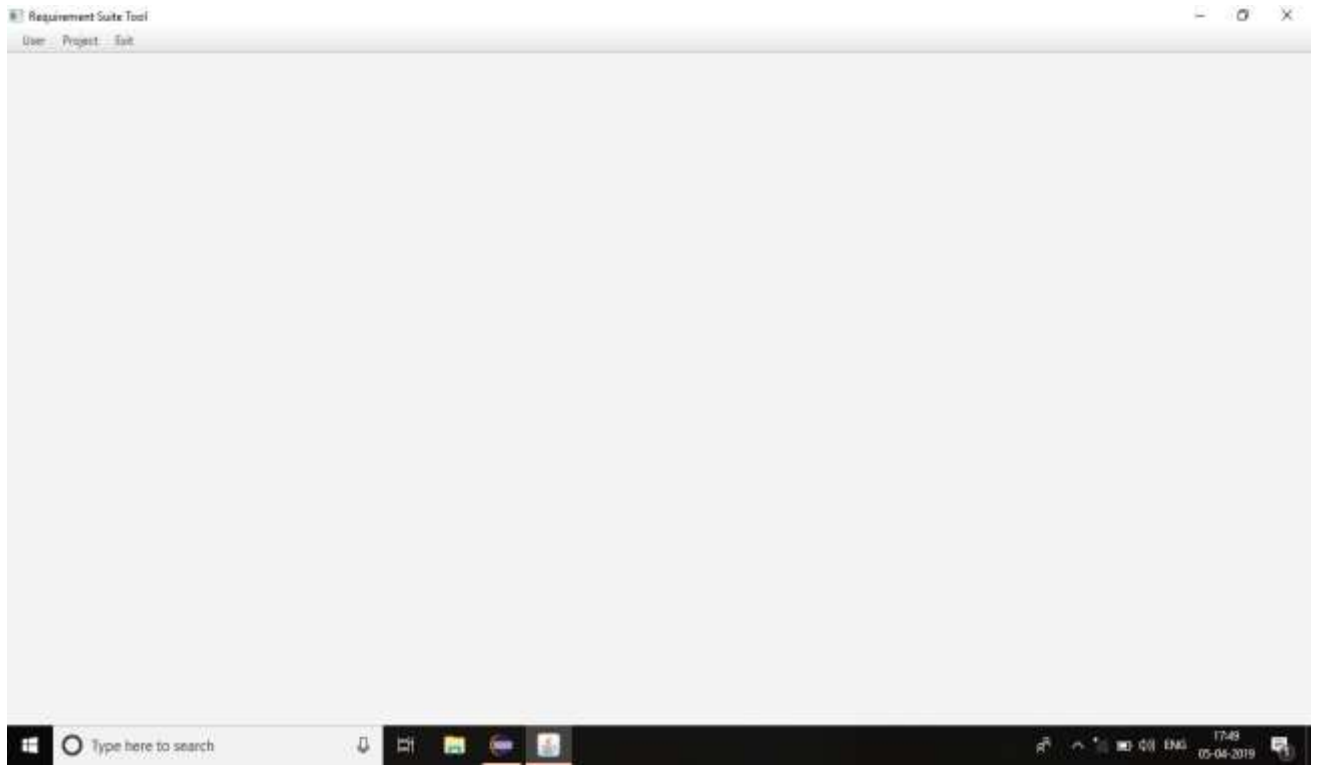
- 1) Industrial Purpose
- 2) Tracing and managing multiple projects
- 3) Medical field
- 4) For beginners to manage and trace requirements.

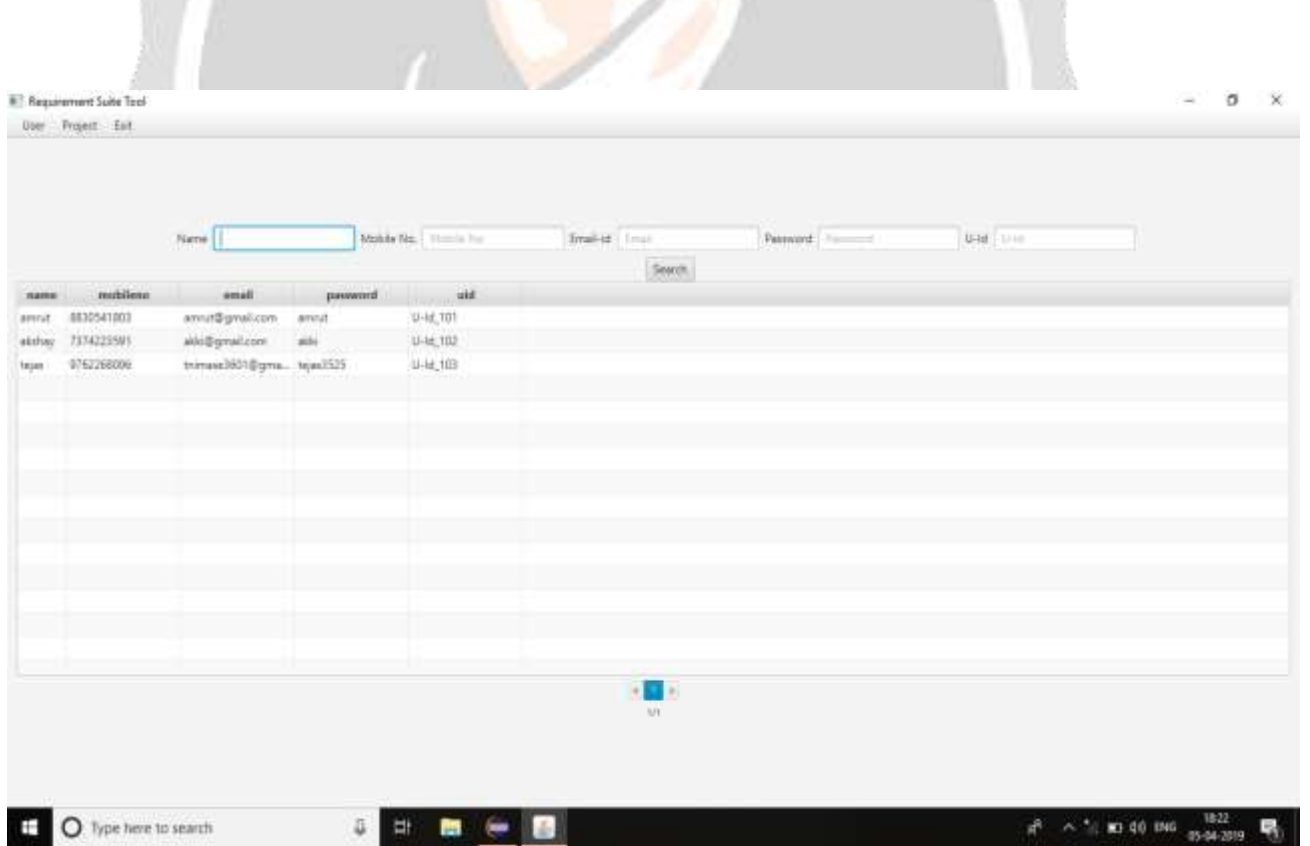
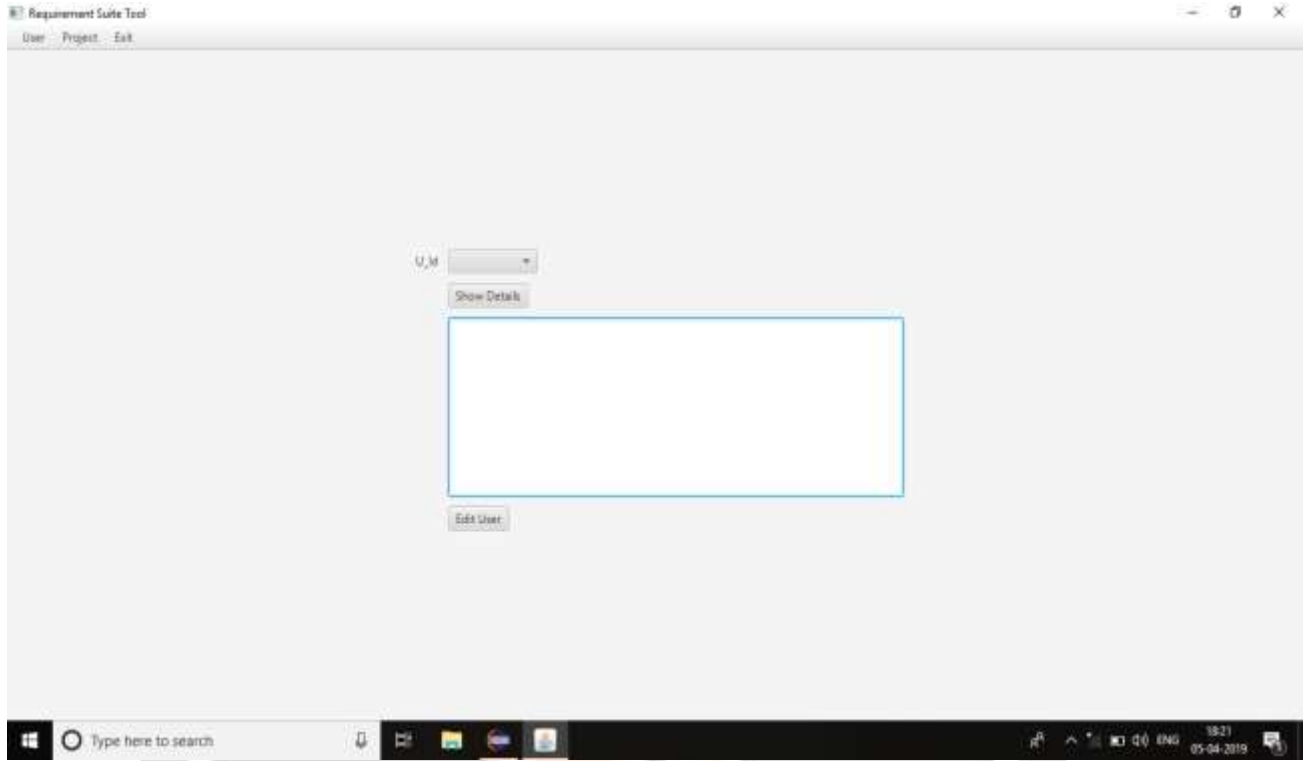
VIII. SCREENSHOTS(OUTPUTS)

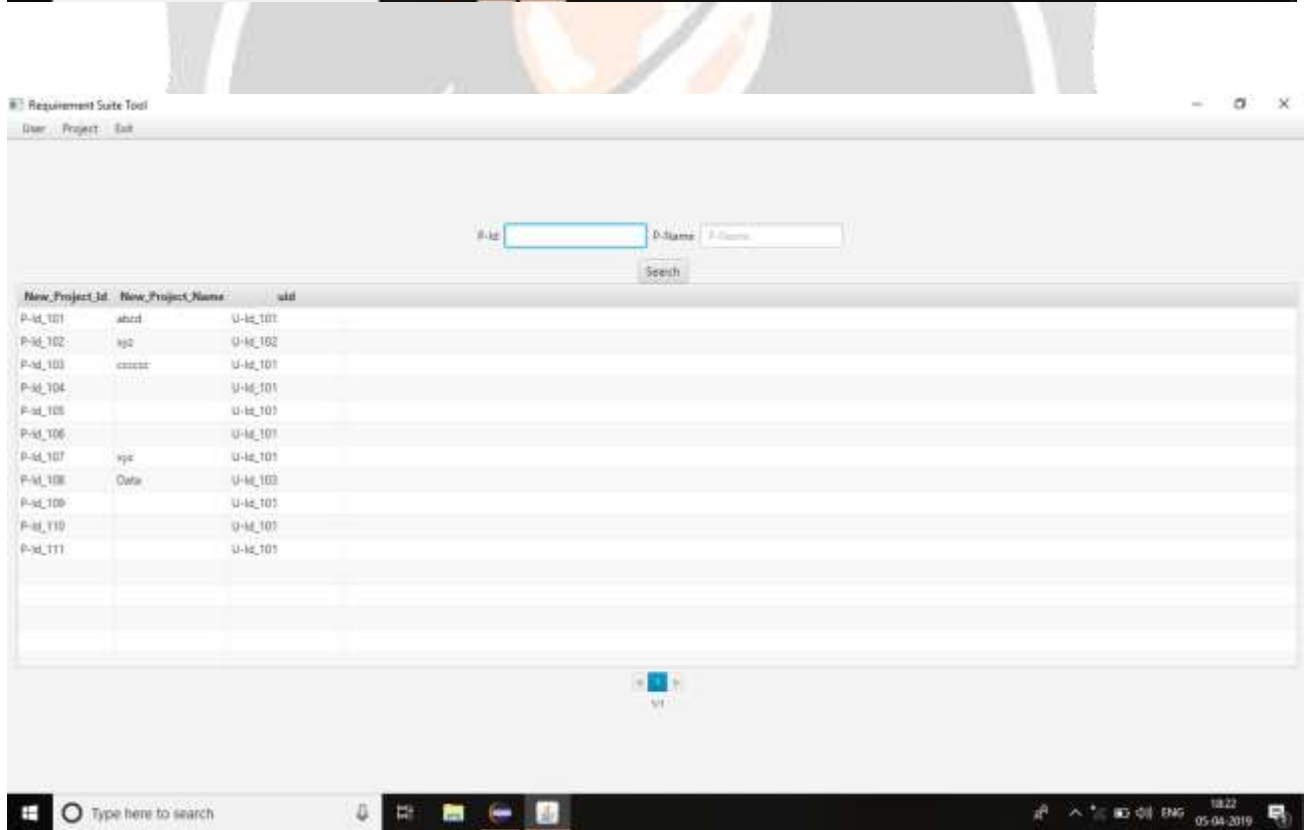
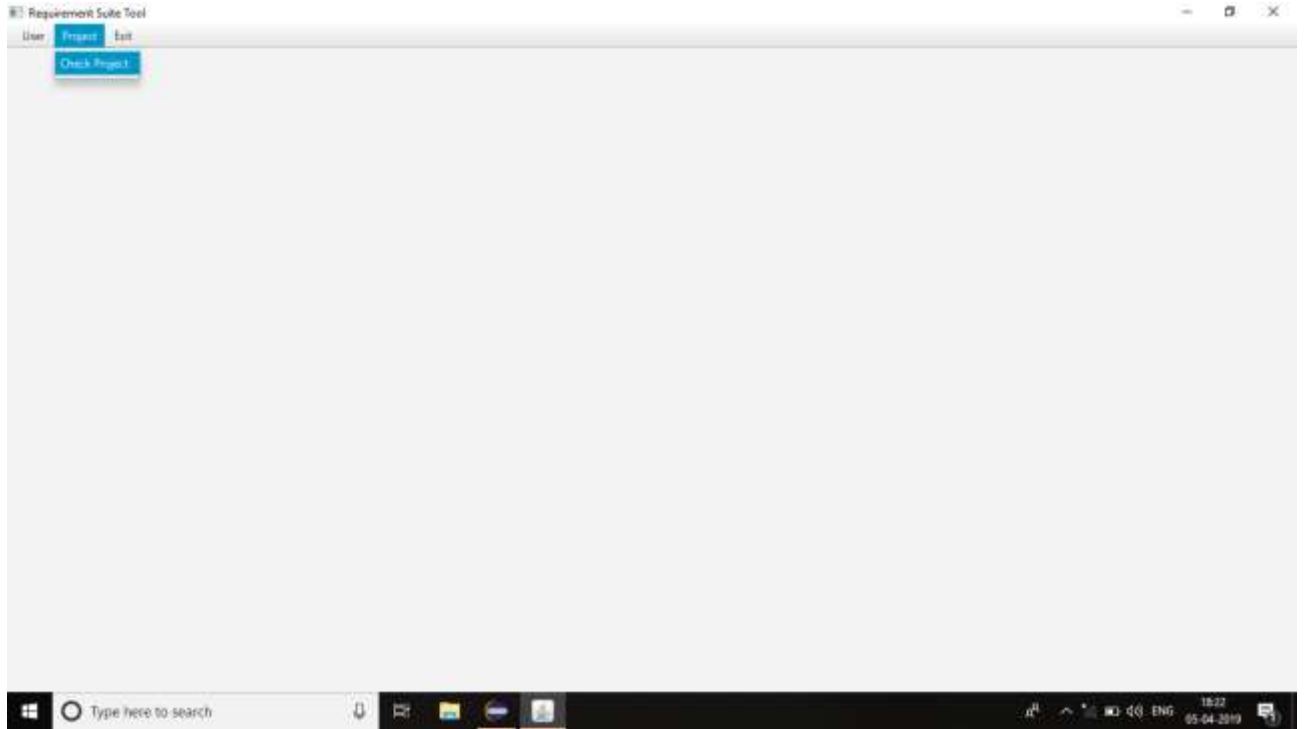
Our Project Contains Two Sides: (A) Admin Side & (B) User Side

(A) Screenshots(Output) From Admin Side:

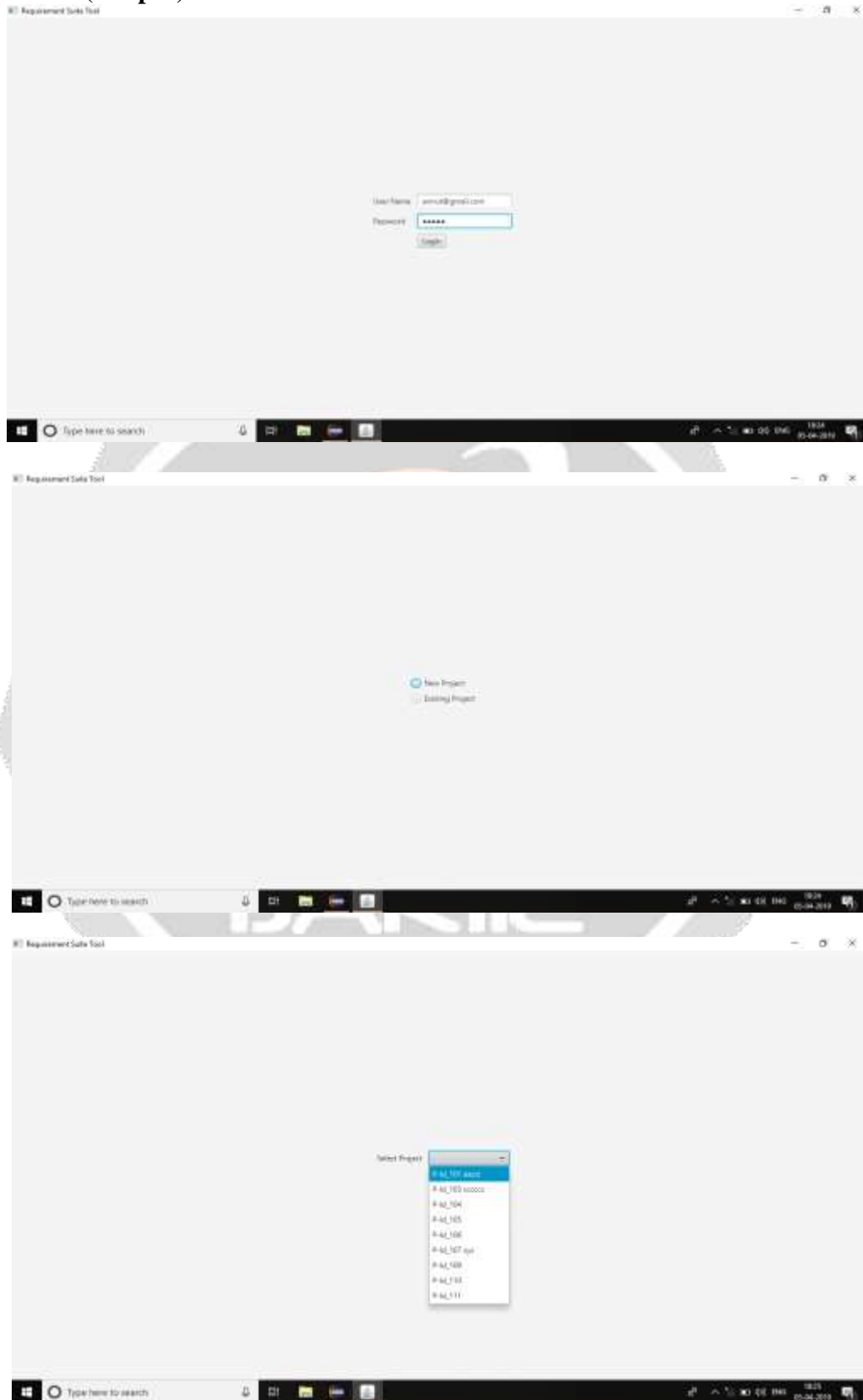


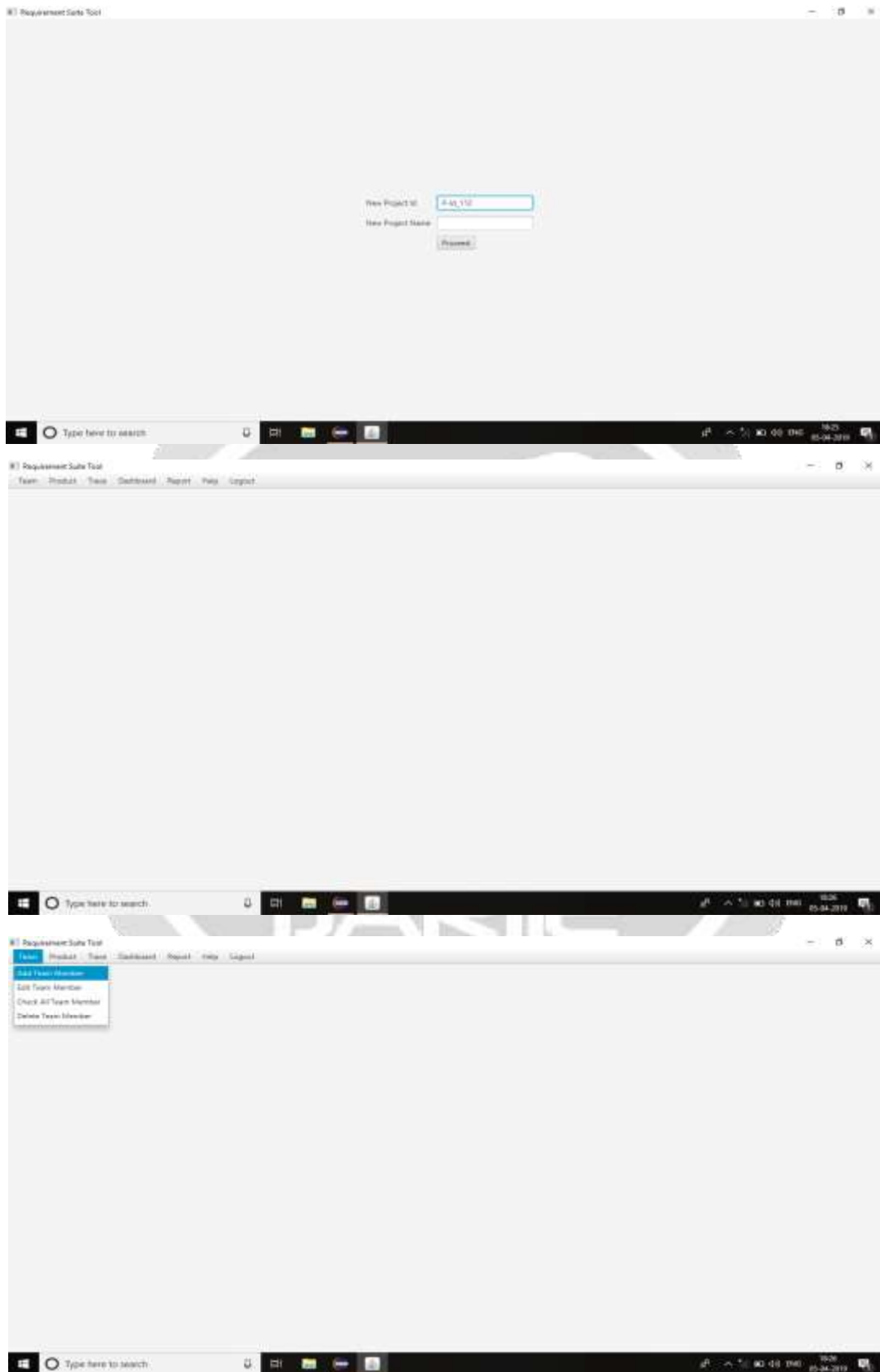


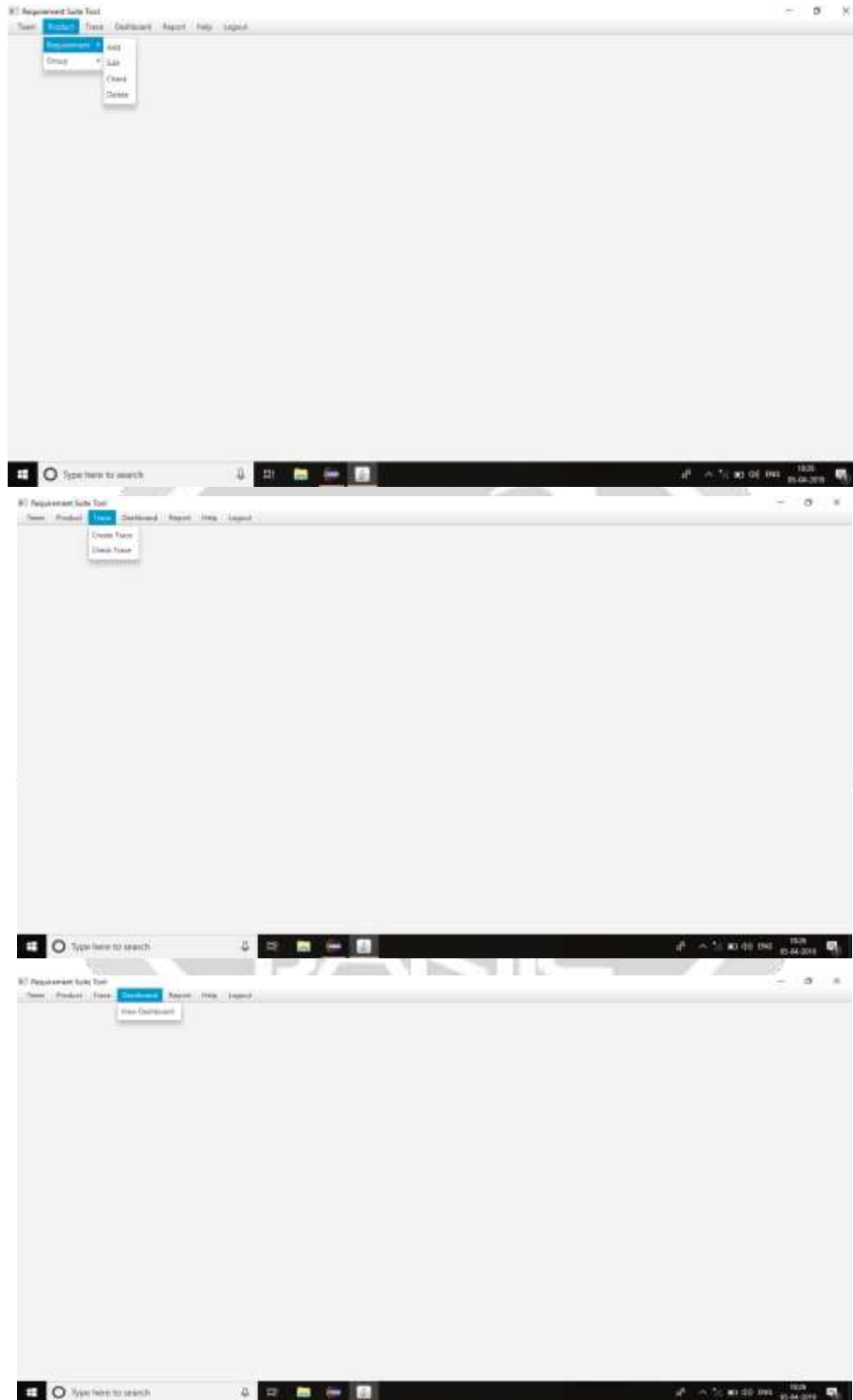


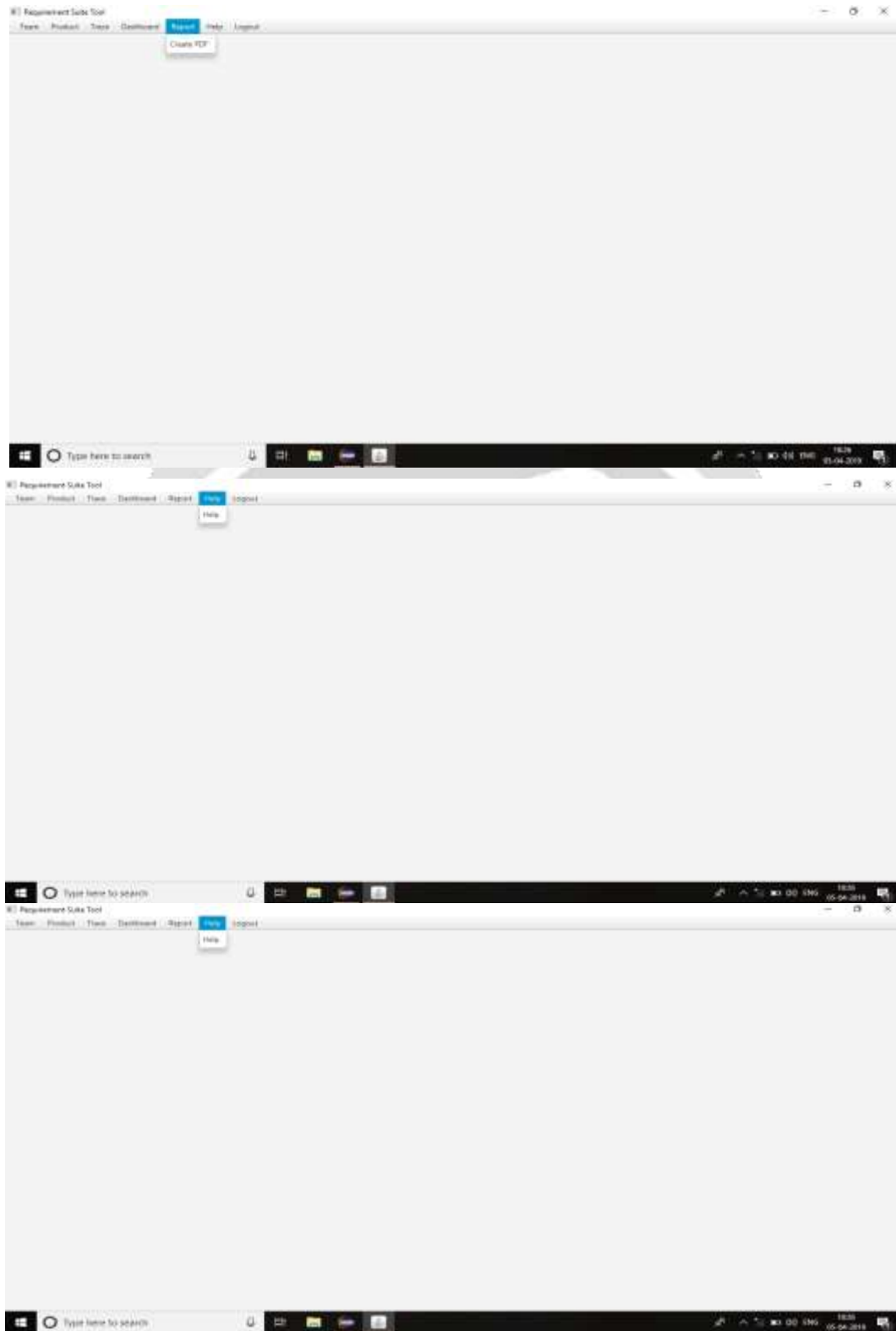


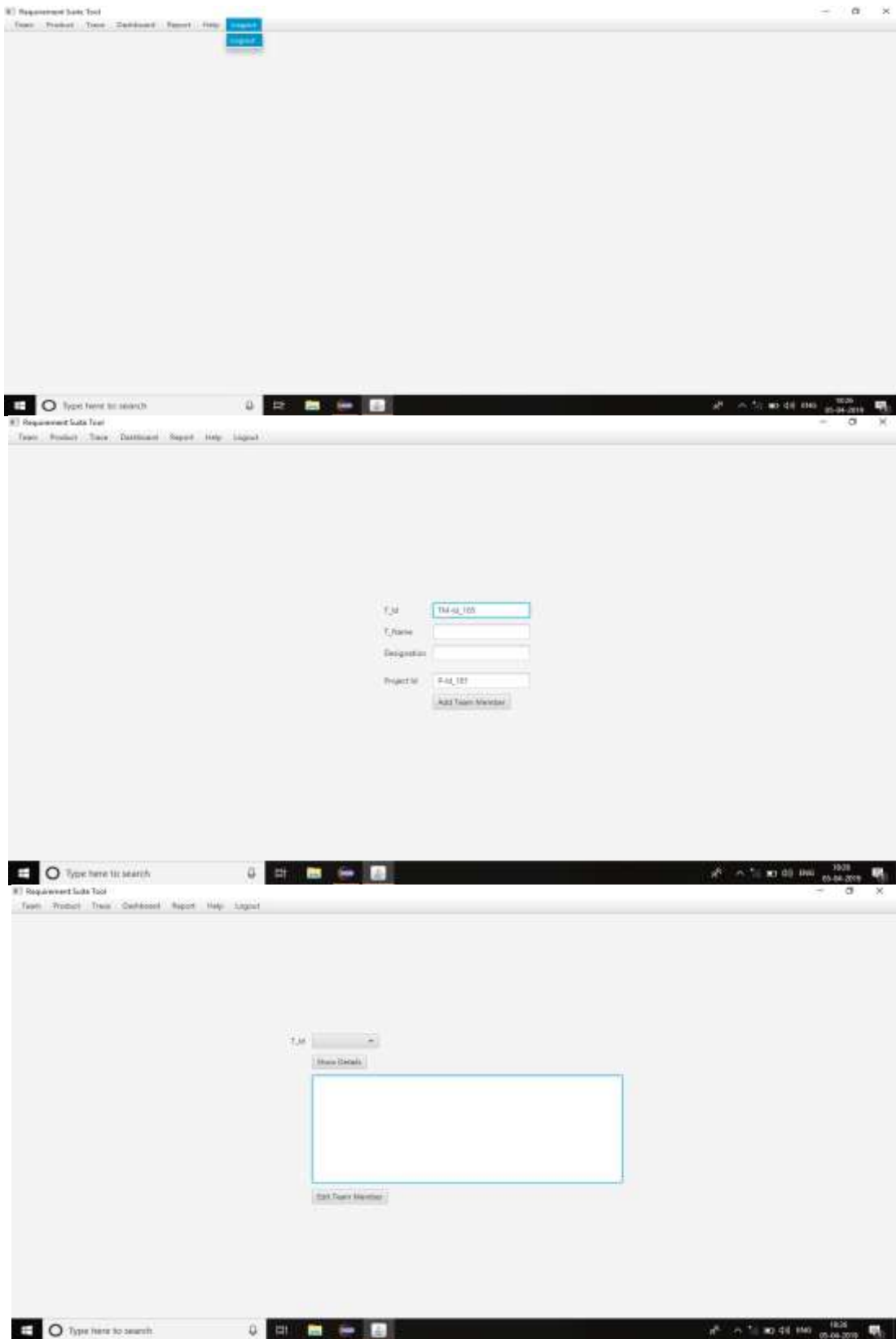
(B) Screenshots(Output) From User Side:

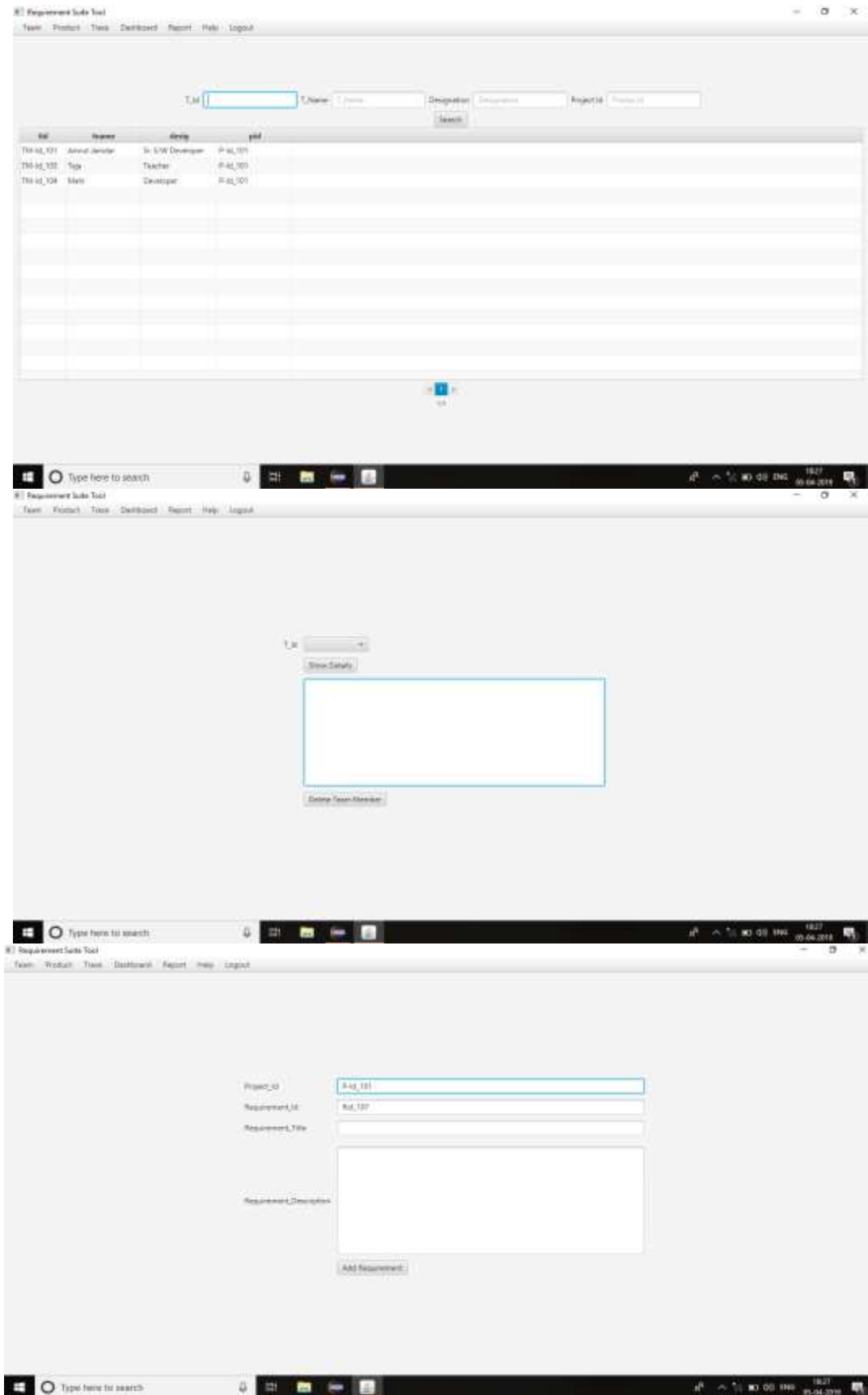


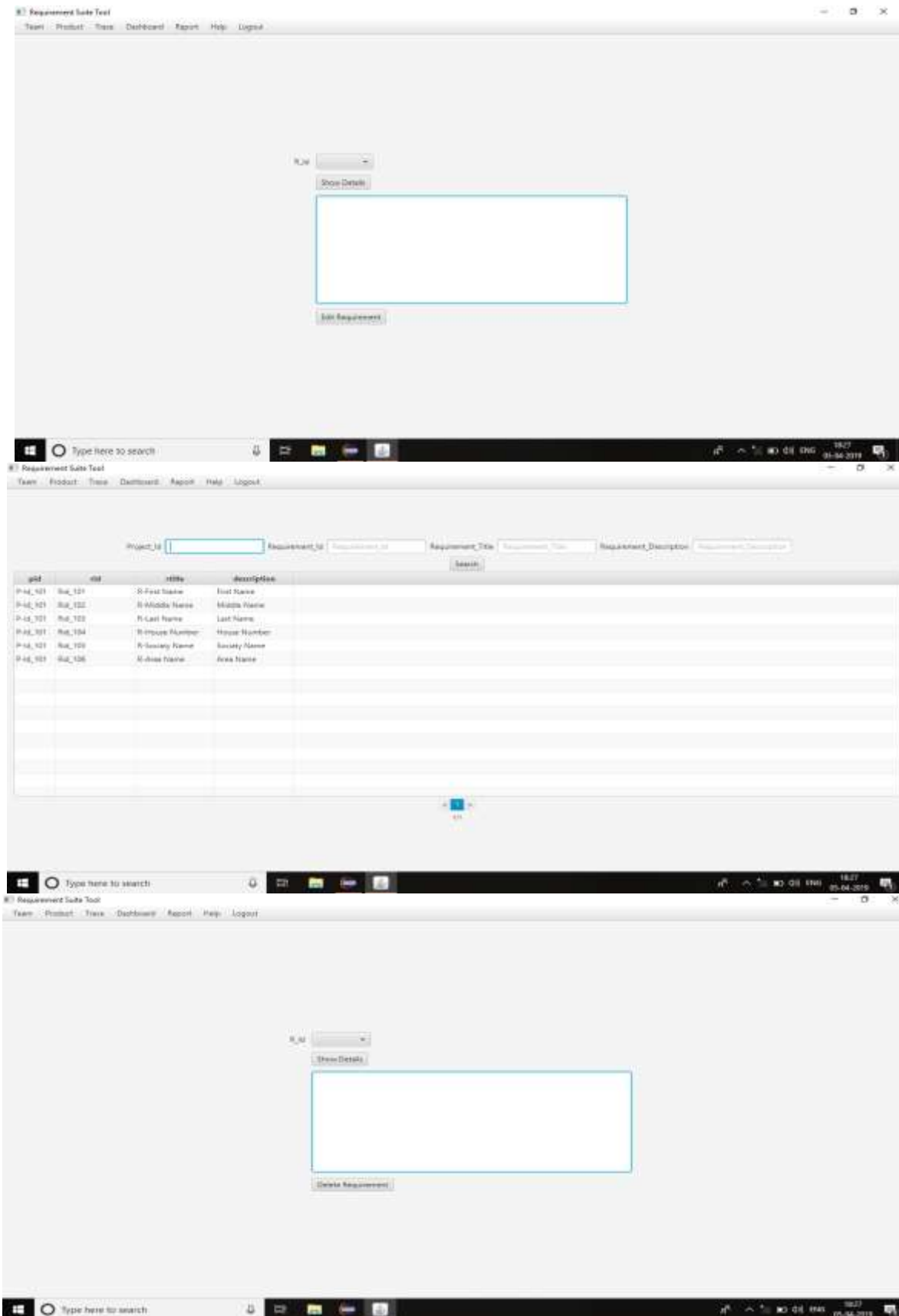


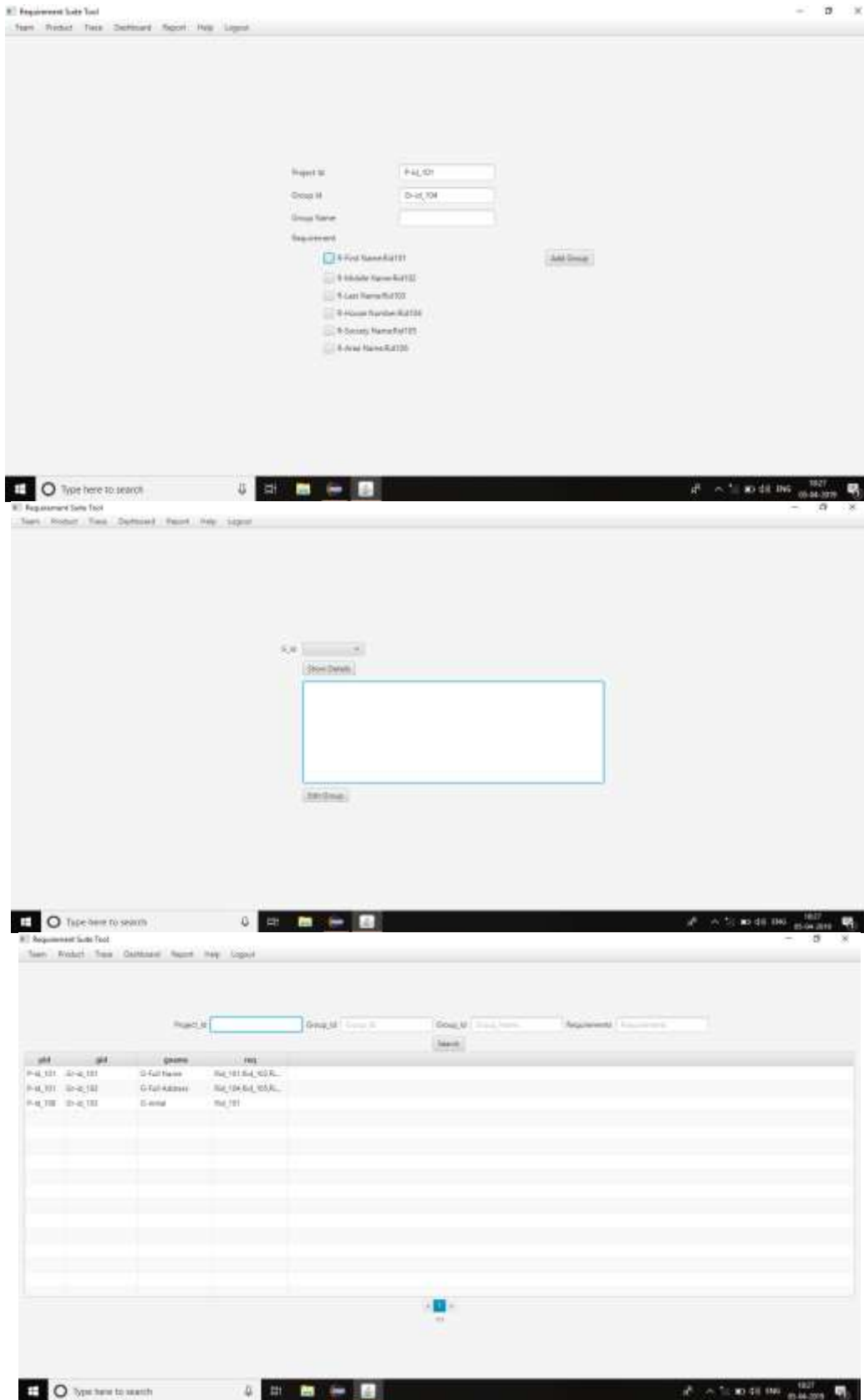


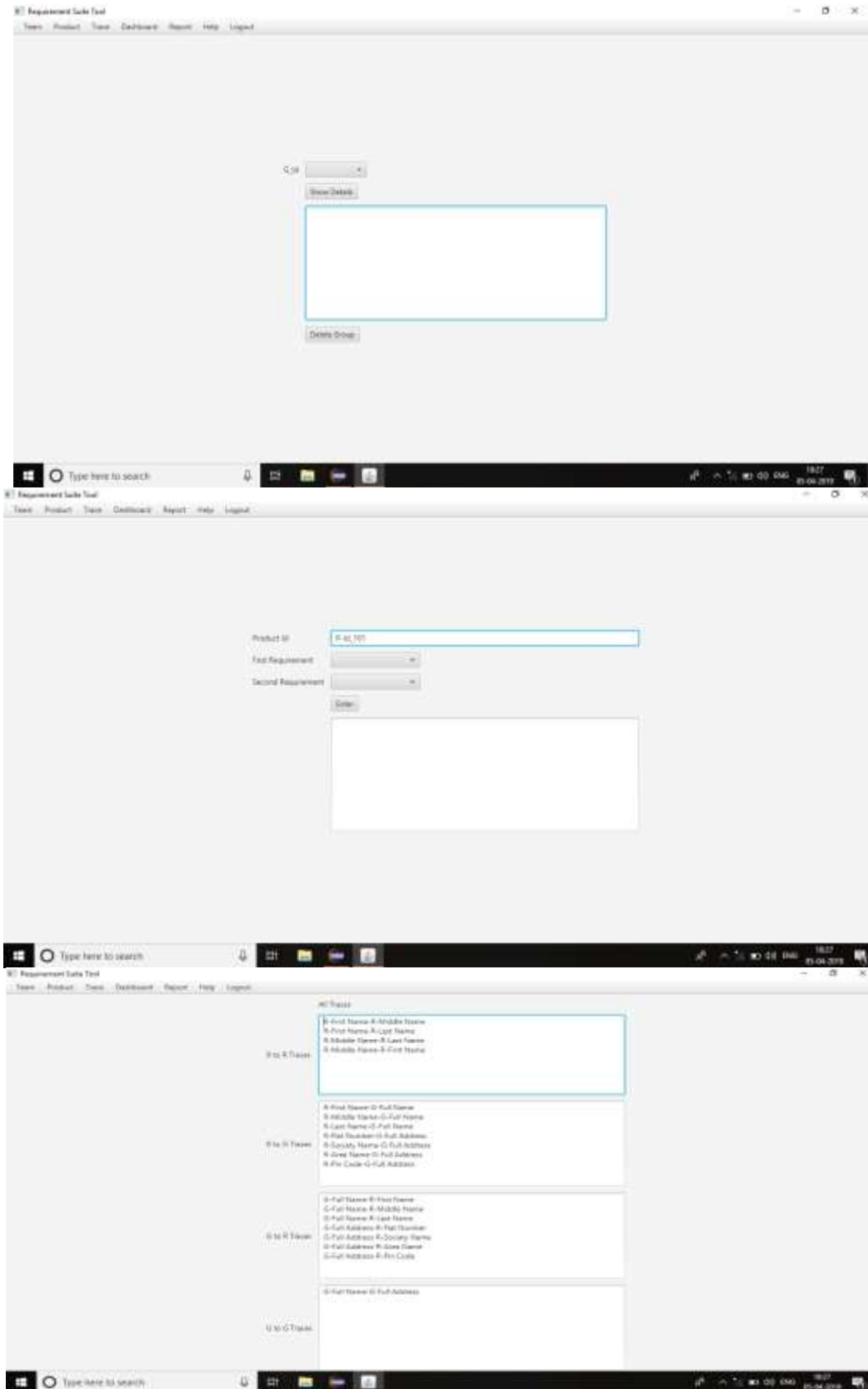


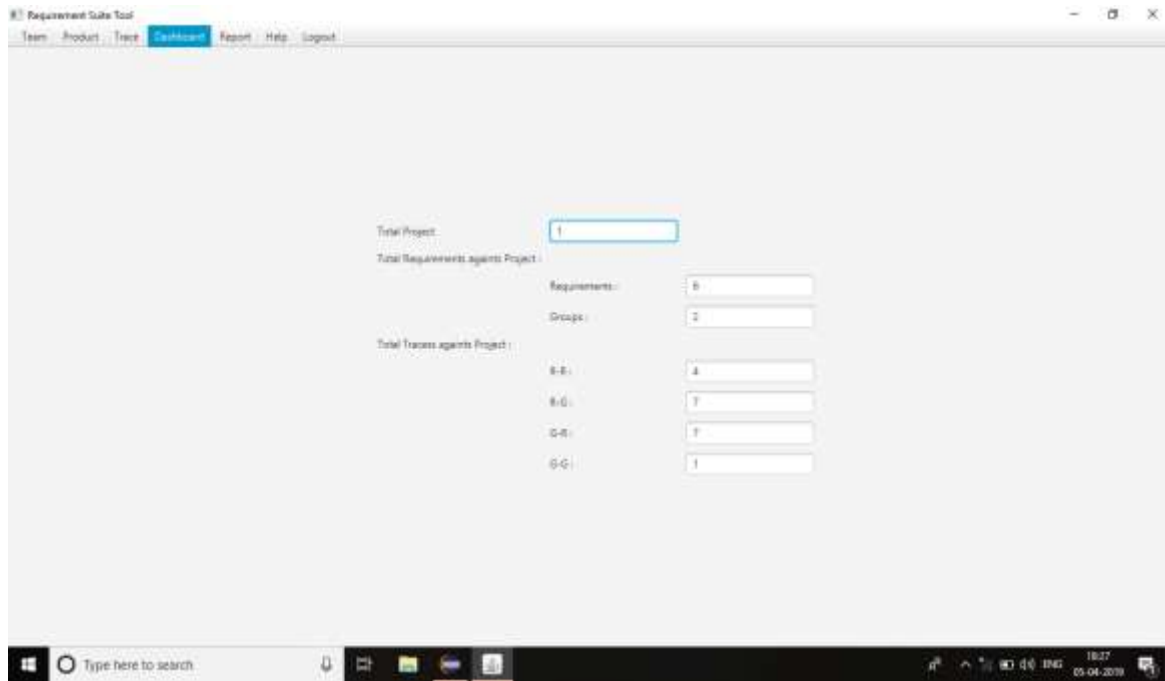












IX. CONCLUSION

Here this Software Development process overcomes all the drawbacks of Requirement Management tool by using Traceability and generates there reports in the form of PDF.

X. REERENCES

- [1] 2010 Second International Conference on Computer Engineering and Applications - “Knowledge Based Requirement Engineering Framework for Emergency Management System”
- [2] 2010 Eighth ACIS International Conference on Software Engineering Research, Management and Applications. – “Requirements Management Tool with Evolving Traceability for Heterogeneous Artifacts in the Entire Life Cycle”.
- [3] 2016 IEEE 24th International Requirements Engineering Conference Workshops. – “Requirements Analysis and Management for Benefiting Openness”