

# **Introduction**

The main purpose of this report is to build a Project Tracker software solution. A Project Tracker software is a system which tracks all the details of the projects, all the tasks in a project, all the users assigned to the tasks and projects, all the project managers assigned to the projects and the users who are not assigned and are currently free. To find a perfect solution for keeping all the functionality of the system and to make it more flexible, following diagrams of the system will be made which are listed below.

1. Use Case Diagram
2. Use Case Documentation
3. Class Diagram
4. Activity Diagram
5. Entity Relation Diagram

# **Use Case Diagram**

A use case diagram showing all the relevant use cases and actors of the system has been shown below using UML notations. UML stands for Unified Modeling Language and it is a popular diagrammatic notation used for visualizing, constructing, specifying and documenting the components of software or non-software systems (Paradigm, n.d.).

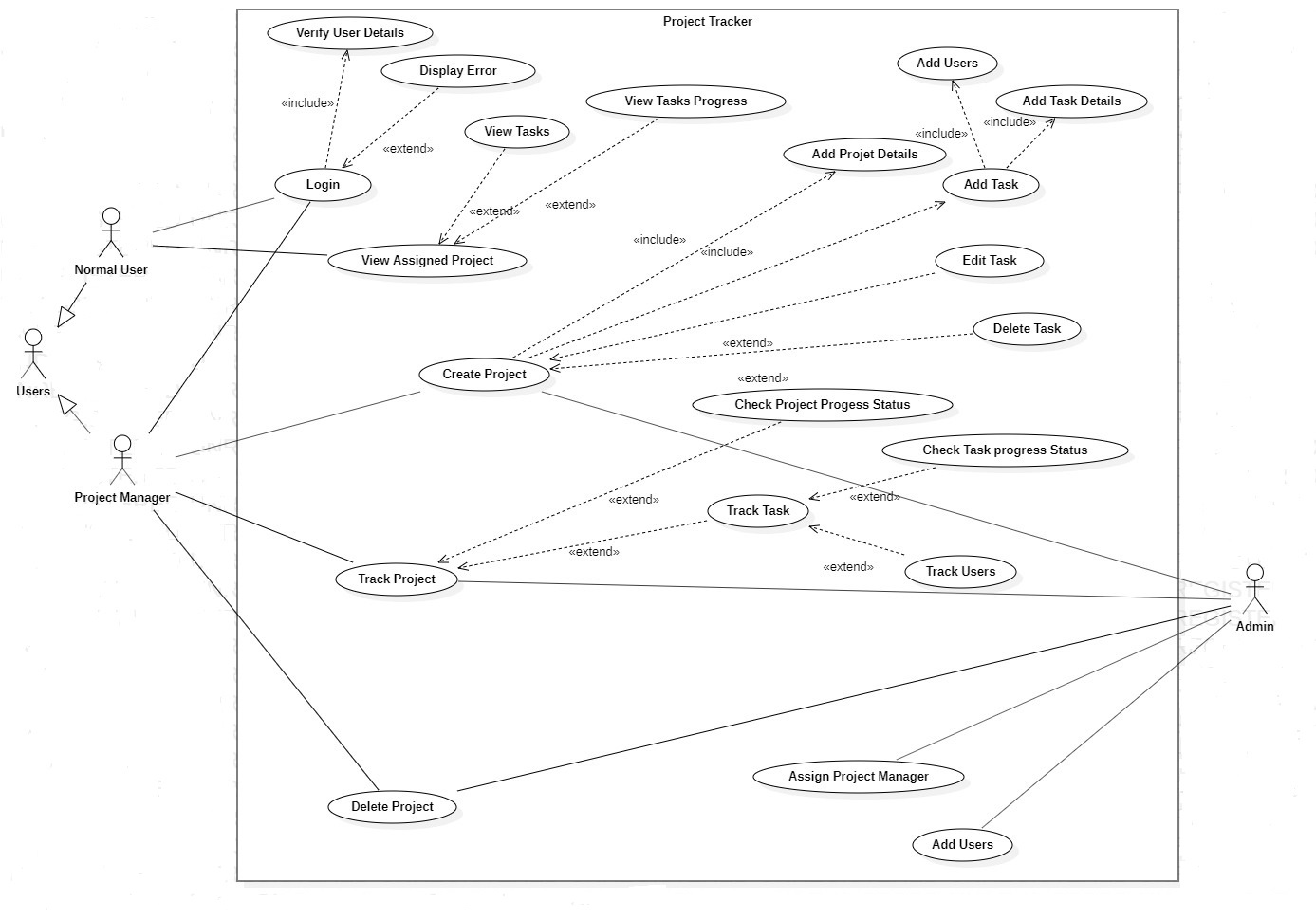


Figure 1: Use Case Diagram

# **Use Case Documentation**

A use case documentation for creating a task by the project manager has been shown below. 

# **Class Diagram**

From the natural language analysis of the given scenario, some nouns and verbs has been identified which are then filtered and possible classes, attributes and methods for the system has been identified. This process has been shown below in a table.



The class diagram for the possible classes, attributes and methods has been shown below.

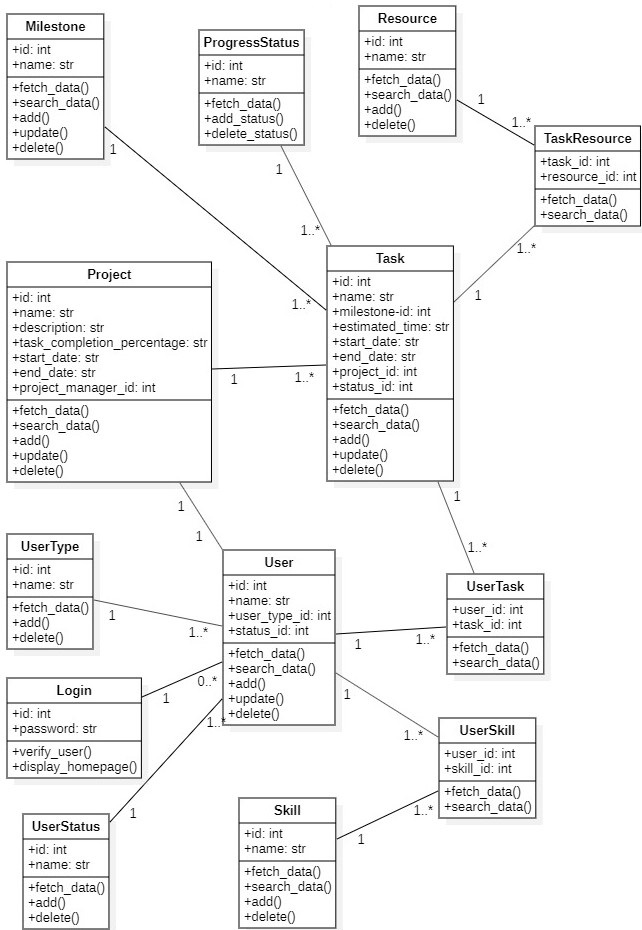


Figure 2: Class Diagram

# **Activity Diagram**

An activity diagram for finding a user in the system, who is not yet assigned to any task and is currently free has been shown below along with the steps.

**Steps:**

1. The user opens the login system
2. User logins with his account
3. System verifies the user’s account
4. System displays the user homepage if his account is verified or displays error if not verified
5. User opens the user page
6. System displays the users list along with the details
7. User searches users by the user status id
8. System verifies the user status id and if the id does not exist, then system displays error. If the id exists but does not matches with the required id, then system displays the users who are assigned. Finally, if the id matched with the required id, then system displays the users who are currently free.

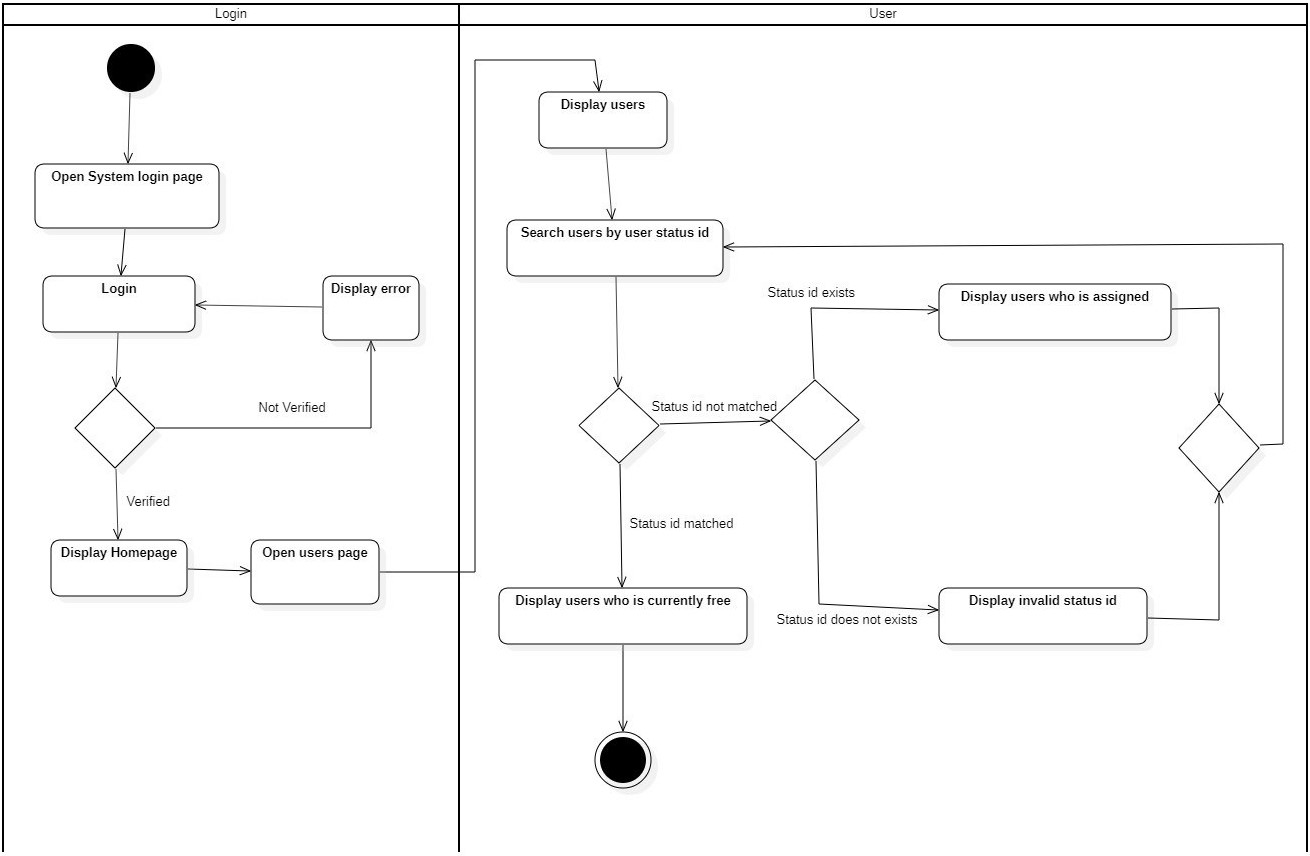


Figure 3: Activity Diagram

# **Entity Relation Diagram**

An entity relationship diagram for this system to identify all relevant entities, attributes, suitable attribute types, primary keys, foreign keys, and cardinalities between entities has been shown below along with all the steps taken to come up with this 3rd form normalized ER diagram using top down approach. Normalization is a method of organizing the tables data in a specific way to reduce data redundancy and insertion, deletion, update anomalies. 1NF, 2NF, 3NF and BCNF are the most commonly used normalization. 3FN table is a table which is in 2FN and [transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) is removed (Singh, n.d.).

**Step 1: Tables for the system (Not normalized)**

* Users table (columns: id, name, skill, user type, user status)
* Tasks table (columns: id, name, users, estimated time, milestone, resource, start date, end date, status)
* Projects table (columns: id, name, description, users, tasks, milestone, task completion percentage, start date, end date)

**Step 2: Tables for the system (Normalized)**

1. A user might be normal at first and then assigned to project manager later. A user also might be free at first and then assigned to the tasks and projects later. Therefore, User table is divided into following tables.

* User Status (id, name), User Type (id, name), Skill (id, name)
* User Skill (user id, skill id)
* User (id, name, user type id, status id)

1. A task might be distributed to multiple users and a project can have multiple tasks and multiple users. A task might be incomplete at first and then might be completed by the users later. Therefore, Task and Project table is divided into following tables.

* Resource (id, name), Milestone (id, name), Progress Status (id, name)
* Task Resource (task id, resource id)
* Task (columns: id, name, milestone id, estimated time, start date, end date, project id, status id), User Task (user id, task id)
* Project (columns: id, name, description, task completion status, start date, end date, project manager id)

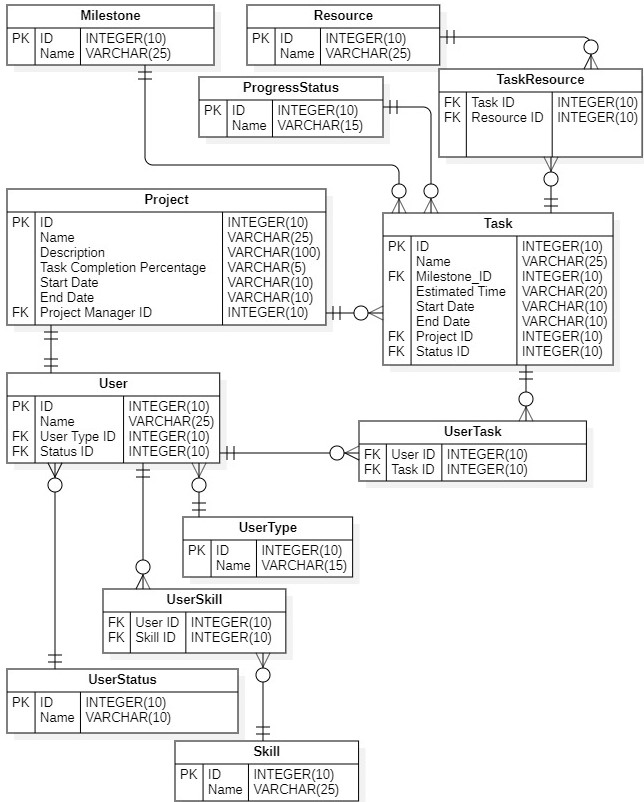


Figure 4: ER Diagram

# **Conclusion**

A perfect solution for the Project Tracker software has been found with the help of Use Case diagram, Use Case Documentation, Class diagram, Activity diagram and ER diagram. Therefore, the system will have all its functionality and will be more flexible to operate without having any problem.

# **References**

Paradigm, V., n.d. *Visual Paradigm.* [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/  
[Accessed 14 August 2020].

Singh, C., n.d. *Begginners Book.* [Online]   
Available at: https://beginnersbook.com/2015/05/normalization-in-dbms/  
[Accessed 14 August 2020].