Introduction:

Welcome to the Software Requirements Specification(SRS) for TimeWise, a productivity management tool through task management. This document highlights the functionalities and requirements of the tool. This document will firstly begin by addressing the inception and elicitation report of TimeWise.Then it will address different modelings like scenario based modelings, database modeling, class based modeling , behavioral modeling . Then finally this document will address the data flow diagram.

Purpose:

The main purpose of TimeWise is to help users manage their tasks in a proper way to become more productive. In this era of social media distractions a tool that organizes daily tasks with proper timings and deadlines along with time to time remainders this tool can lessen procrastination and can make daily activities more manageable.

Intended Audience:

All the reports,modeling and diagrams used in this diagram can give a prosper and clear idea about our tool which can be helpful for software developers,stakeholders,system architects, project managers, quality assurance professionals,end users as well as someone who is interested about the functioning of TimeWise.

Conclusion:

In conclusion we the creators of the tool Abdus Salam Islam Badhon ([bsse1401@iit.du.ac.bd](mailto:bsse1401@iit.du.ac.bd)) and Md Nahidur Rahman Nahid ([bsse1429@iit.du.ac.bd](mailto:bsse1429@iit.du.ac.bd)) believe that this SRS will help the intended audience to get the clear understanding of our tool. And we are also grateful to our supervisor Mridha Md Nafis Fuad([fuad@iit.du.ac.bd](mailto:fuad@iit.du.ac.bd)) sir for guiding us regarding the project.

1. Inception Report:

1.1 Project Overview and Objectives

TimeWise is a productivity tool that helps users to manage their daily activities. For now we are only concerned with making a web app to provide the service. Later we may consider other kinds of services like mobile apps ,desktop apps or for some other platforms. And out site is for now totally free to use so this document will not address any payment related specifications.TimeWise will have different features like listing daily activities in a to-do format, tracking progress and performance, getting time to time reminders and notifications, seeing relevant statistics of the participations,getting personalized feedback and tips and managing productivity in a collaborative and competitive environment.

1.2 Stakeholders

TimeWise is a tool which can be useful for anyone trying to organize their task and time. It is not created based on a particular group of people like students, working professionals,employees,managers and so on. It can be useful for all of the categories of people. So in the rest of the documents we will address all of these categories of people simply as users.Along with these stakeholders there will be admins who will manage the tool and users activities.SO the stakeholders are the followings:

1.Users

1.1 Students

1.2 Teachers

1.3 Working Professionals

1.4 Employees

1.5 Managers

1.6 Others

2. Admins

1.3 Methodology and Requirements Collection Plan

To collect requirements we will talk to the stakeholders personally and will collect their personal experience and recommendations regarding managing their daily activities. We may also collect information from a vast amount of stakeholders by some online means. We will also refer to research papers, reports,relevant books for collecting information about productivity related contents. We will take notes about our findings and will refer to these in developing and planning the system.

1.4 Assumptions and Risk

We will assume that the user will behave in a fair way like giving the actual input. If they dont the outcomes and results that our system will show will be biased and will be not beneficial for the users. Our current system will not address any dedicated efforts to stop cheating of the user. Maybe in the later versions we may consider these issues. So we expect the user to be fair for their own benefits.

1.5. Timeline and Strategy

In the beginning phase we will be collecting requirements and relevant information and make proper documentation of the findings. Then the UI/UX planning and actual development will be done. Followed by testing and bug fixing. Finally we will deploy the tools and will provide user documentation for better user experience. We will follow the iterative process model to develop the actual system. Where after each iteration we will perform the required changes. And we may develop independent part parallel so that integration of different modules becomes easy.

2. Elicitation Report:

2.1 Quality Function Deployment(QFD)

The Quality Function Deployment(QFD) is a structured approach to define user needs or requirements and translating them into specific plans to produce products to meet those needs.Based on our observations after interaction and research on the subject we summarized the following requirements that will be implemented on our tool.

2.1.1 Normal Requirements

**1. User management:** The user will be able to create an account by providing some personal information like name,email address and so on and our system will maintain a profile for each user.

**2. Task Management:** The user will maintain a to-do list which will contain the task that the user feels to be done in short or long terms.

**3. Progress Tracking:** Based on setted goals, the user will be able to track his progress of the current assigned tasks.

**4. Performance Tracking:** Users can evaluate his performance based on the completed tasks.

2.1.2 Expected Requirements

**1. Reminders & Notifications:**Users will be notified from time to time based on his setted tasks and activities.

**2. Showing Stats:** Users will be able to see relevant statistics about his performance and participation.

2.1.3 Exciting Requirements

**1. Deep work tracking :** Users will be able to see in which time span users can perform better.

**2. Collaborative and competitive environment:** Users will be able to share their goals and plans with others. They can evaluate their performance with respect to others.

3. Scenario Based Modeling

The success of a computer-based system depends highly on the user's satisfaction. So if we can create different scenarios of users interacting with our system we can target these interaction scenarios and make them useful to give an overall good experience. Scenario based modeling highlights these users' interactions. This modeling uses different diagrams like use case diagram, activity diagram and swimlane diagram.These diagrams along with proper description and explanation are given below:

3.1 Use Case Diagram

A use case is a detailed description of how a user interacts with a system to accomplish a specific goal. It focuses on the user's perspective, outlining the steps involved. Primary actors are the main users directly interacting with the system and positioned at the left side , while secondary actors are external entities that provide or receive information but aren't directly using the system positioned at the right side.In the middle the main modules with which the user interacts are placed.Actors doesnt need to be physical person it can be intangible entity also like mail,third party software and so on.To show the interaction between modules and actors a line is drawn between actors and modules name. Now the use case diagram of TimeWise will be shown below:

3.1.1 Level 0

**Name:** TimeWise(A time and productivity management system)

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Admin,Central Database,Notification System,Analytics Engine

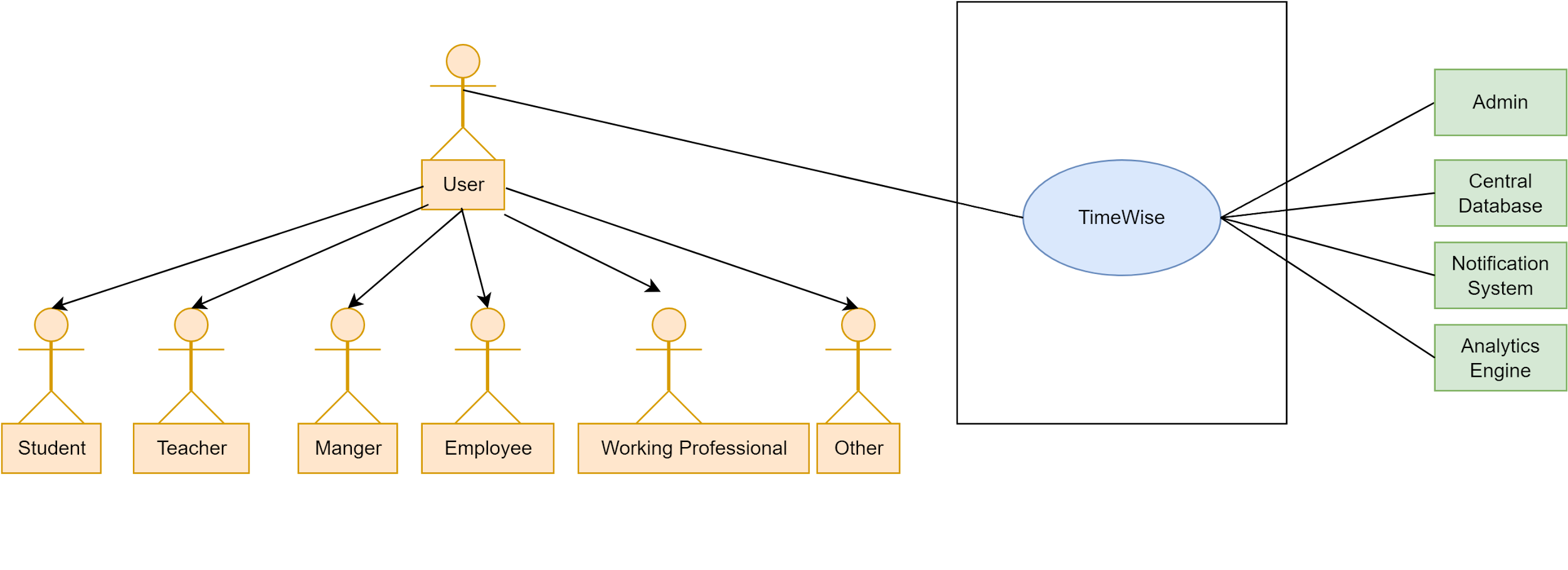
[](https://app.diagrams.net/?page-id=lnonrUPNMiSJG-7Qp96J&scale=auto#G1OXqhkd54wwl-46aviZ00bA-DMpYdVPli)

Figure 1: TimeWise(A time and productivity management system)

3.1.2 Level 1

**Name:** TimeWise System Details

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Admin,Central Database,Notification System,Analytics Engine

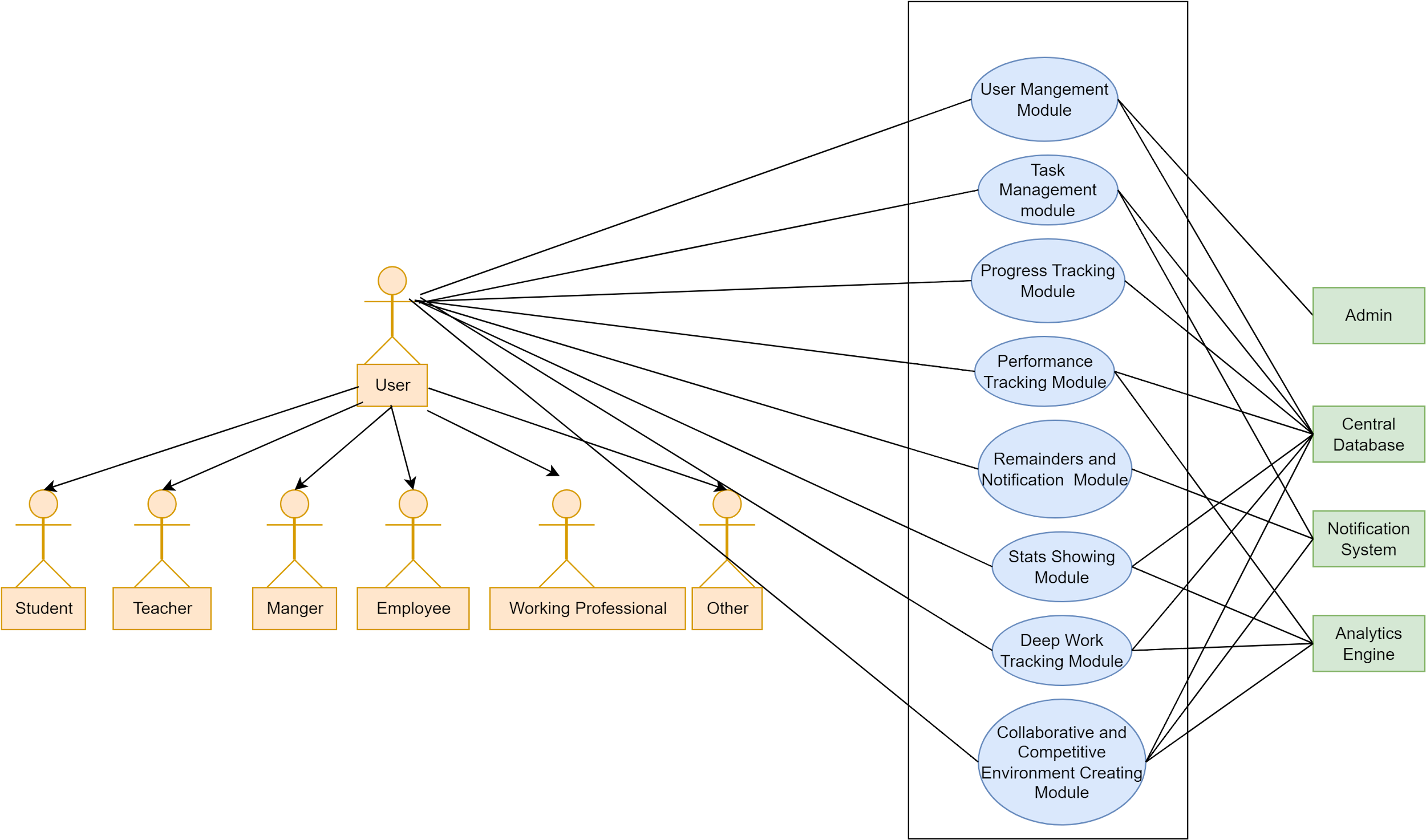
[](https://app.diagrams.net/?page-id=5Ep43n3G_8iIJSV3E2aV&scale=auto#G1ppG80JHn_XJio1zb_8FSoVjxcr_4WeyS)

Figure 2: TimeWise System Details

**Descriptions:**

**User Management Module:** Manage users account related activities like sign in ,sign up,collecting user credentials and maintaining a user profile and dashboard.

**Task Management Module:** Get task information and their corresponding deadlines. This module will contain all the functionalities regarding manipulating and updating the to do list.

**Progress Tracking Module:** Shows the progress of ongoing tasks and relevant information to complete the task.

**Performance Tracking Module:** Shows how the user performed over the period of time on different completed tasks and activities.

**Reminders and Notification Module:** Sends time to time reminders and notifications.

**Stats Showing Module:** Shows relevant statistics regarding the performance and participation of the user.

**Deep Work Tracking Module:** Shows how productive or efficient the users are over the different time spans.

**Collaborative and Competitive Environment Creating Module:** Creates a platform where users can collaborate with each other and become more productive by healthy competition.

The user will interact with all the modules as they are consumers of the whole system. The mail system will be used for sending notifications and reminders. The database will be a key part for providing and storing data to different modules. The admin will take part in managing users accounts.

3.1.2.1 Level 1.1

**Name:** User Management Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Admin,Central Database

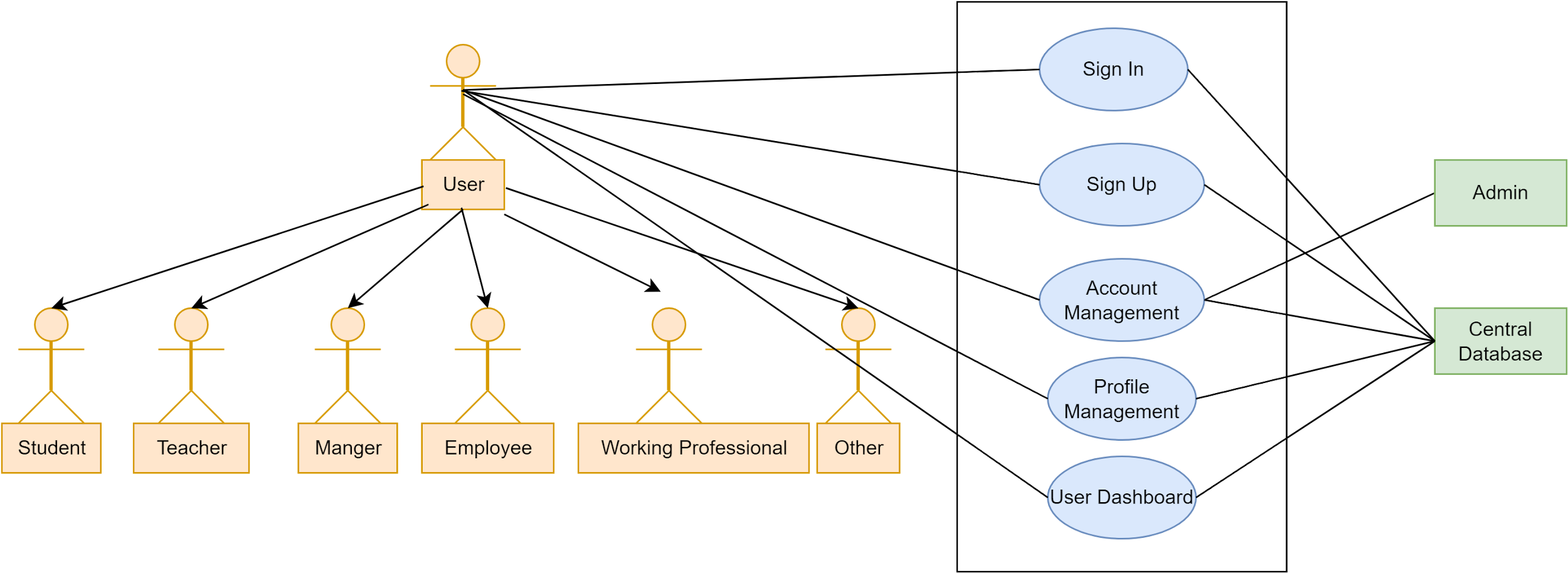
[****](https://app.diagrams.net/?page-id=IOBFKqrxkaWz8YnhxEr4&scale=auto#G12Biwgp09lsqEmQ0w6_07t-LmRrbIcaiM)

Figure 3: User Management Module

**Description:**

Users will be able to sign in and sign up by providing necessary credentials. They can manage their account by deleting or deactivating their account. Admins will also be able to govern the account of the user like removing unlawful users from the system. Users will also be able to modify their profile details and can see all of their information in the user dashboard.

3.1.2.2 Level 1.2

**Name:** Task Management Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Task Scheduler Service,Central Database, Date and Time Service

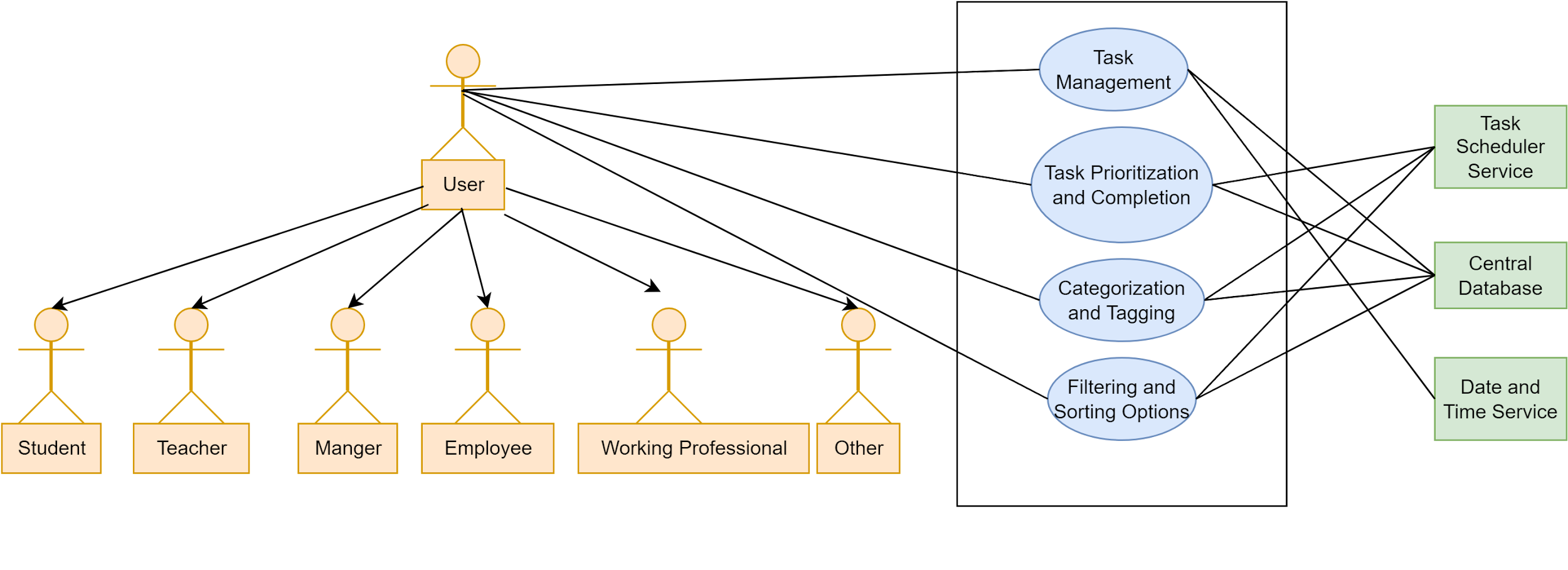
[](https://app.diagrams.net/?page-id=7bbSqT4xshHzTUvxiv2a&scale=auto#G1e71-aSUyi3OHE7HaaMWfx4uTWoyXS94q)

Figure 4:Task Management Module

**Description:**

Users will be able to manage tasks like adding,removing,updating tasks details. They can set and modify the details about task prioritization and completion. They can categorize,filter,sort and tags different tasks.

3.1.2.3 Level 1.3

**Name:** Progress Tracking Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Central Database, Goal Tracking Engine, Progress Analytics

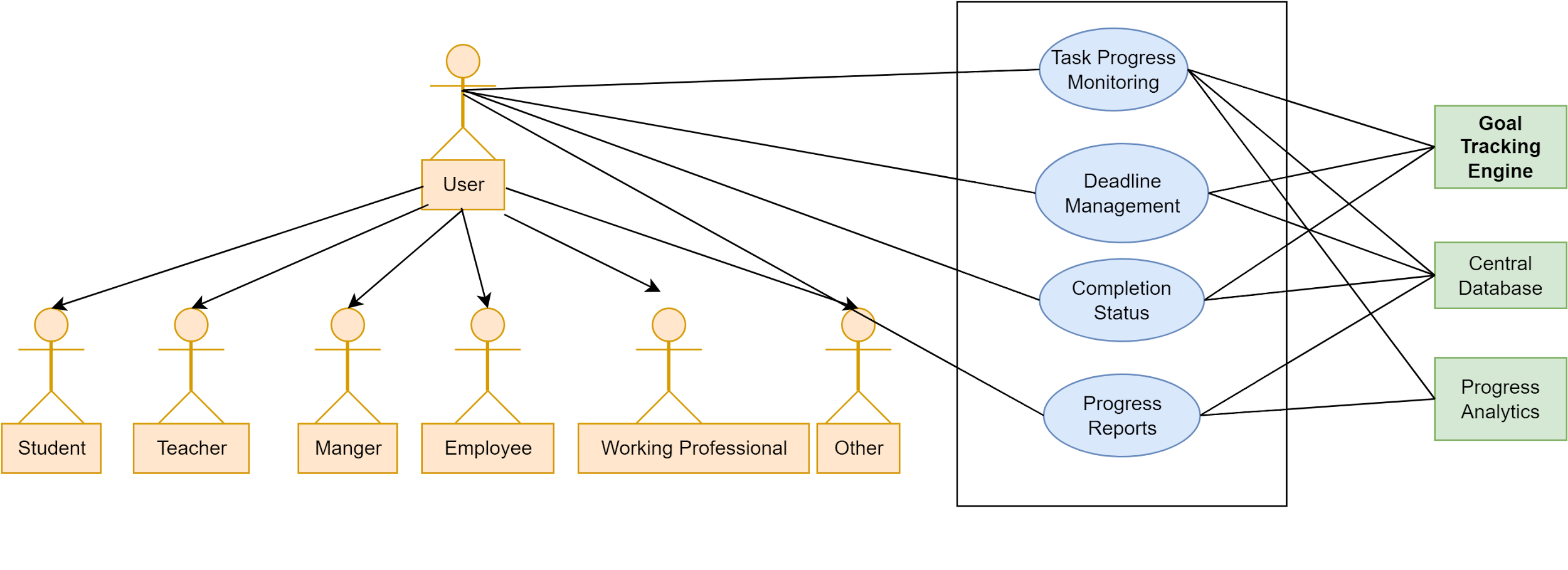
[](https://app.diagrams.net/?page-id=-0hZbVNOpv6FWO8korFR&scale=auto#G1DLqsWB9EAKr0YXlzFwhgBbTT-yNxr5JQ)

Figure 5:Progress Tracking Module

**Description:**

Users can monitor progress status like seeing deadlines and completion status which can help them to focus on the tasks that are close to the deadline but not yet finished. They see the progress reports to work on the ongoing task.

3.1.2.4 Level 1.4

**Name:** Performance Tracking Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Central Database,Performance Analytics Engine, Reporting Service

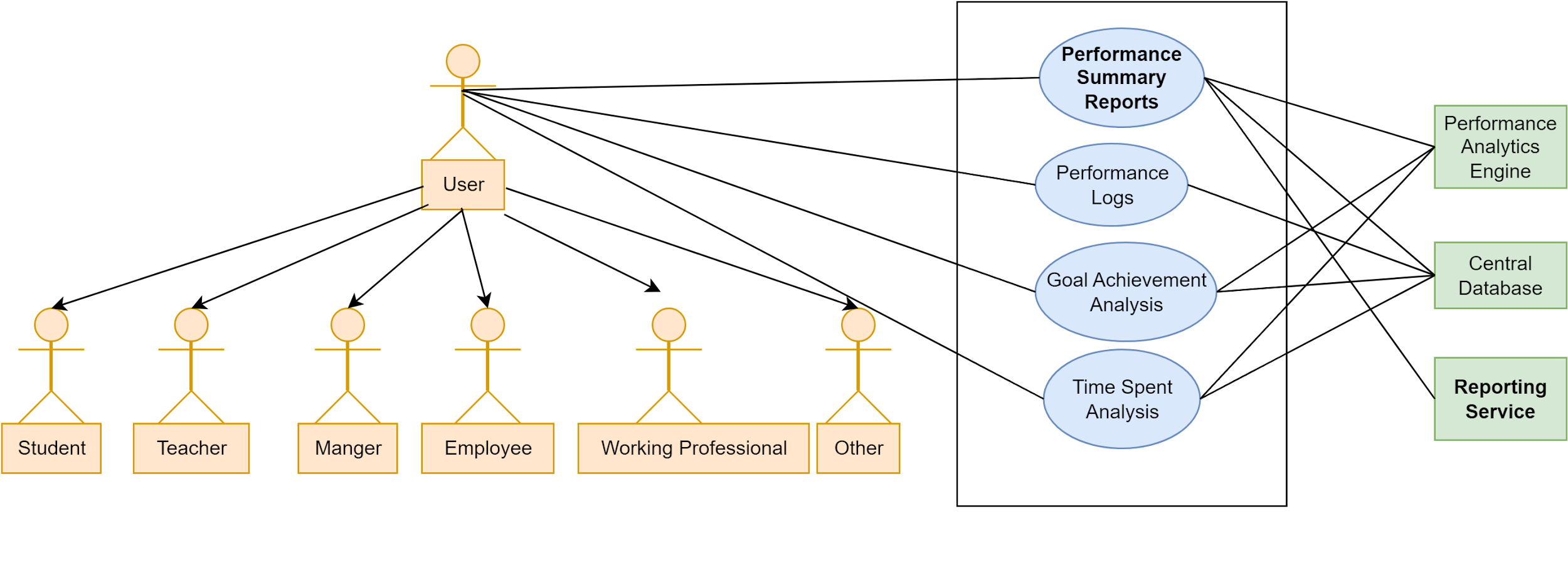
[](https://app.diagrams.net/?page-id=Wfa6qR6buEkdc9ZD8dWT&scale=auto#G1nNRFF8iyLKwNeBe6sPwFETc1vS_HL57e)

Figure 6:Performance Tracking Module

**Description:**

Users can see the performance reports which may be helpful to evaluate his performance. They can see performance logs on different time intervals. They can also see goal achievement as well as time spent analysis.

3.1.2.5 Level 1.5

**Name:** Reminders and Notification Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Central Database,Email Service

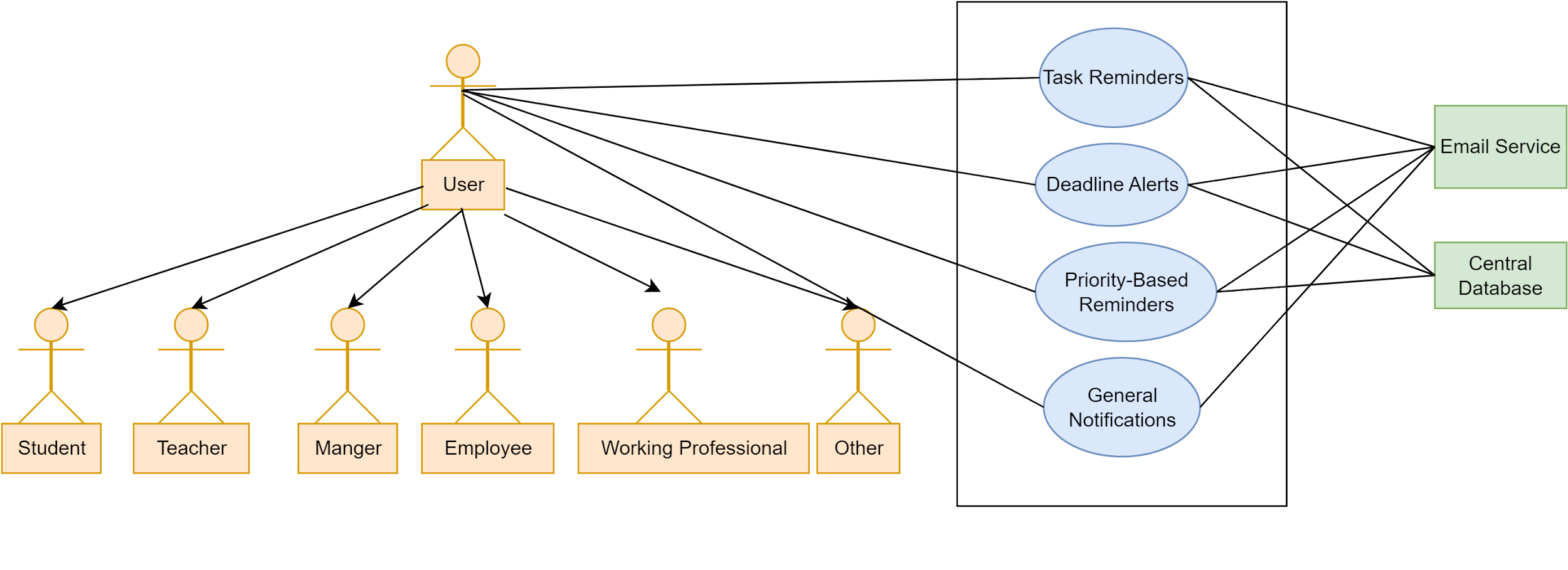
[](https://app.diagrams.net/?page-id=txsZjO7vKs0XT66AJvK2&scale=auto#G19osDktNTyjxZllwFMAZWI1FuzvMWUKe4)

Figure 7: Reminders and Notification Module

**Description:**

Users will get time to time alerts to deadlines and unfinished tasks. They will also get general notification regarding the service. Based on priority of tasks they will also get notification.

3.1.2.6 Level 1.6

**Name:**  Stats Showing Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Central Database,Statistics Engine,Visualization Service

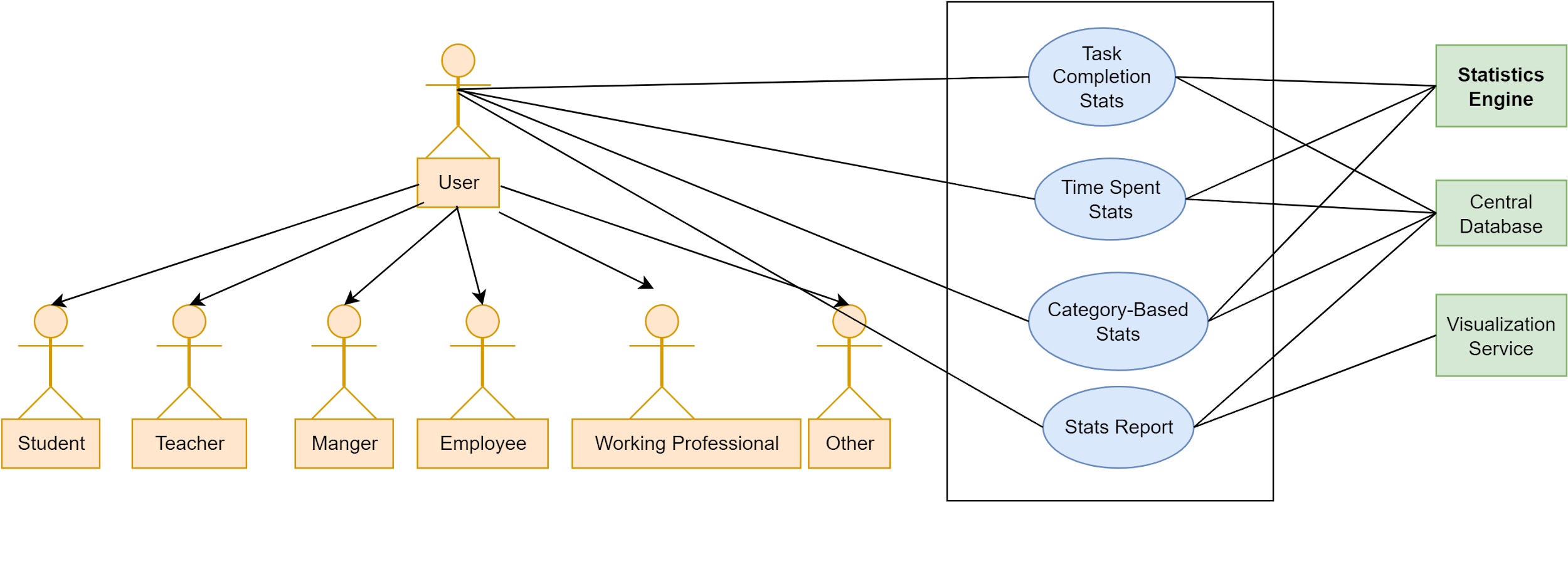
[](https://app.diagrams.net/?page-id=vaSSFvOXgG2yqRM5z1OH&scale=auto#G1mPFCE7wSViQOHLI0ChTtA1Jl45ks3jIS)

Figure 8: Reminders and Notification Module

**Description:**

The statistics engine with the help of a central database will visualize the user performance with the help of relevant statistics.

3.1.2.7 Level 1.7

**Name:**  Deep Work Tracking Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Central Database,Focus Timer Service,Efficiency Analytics

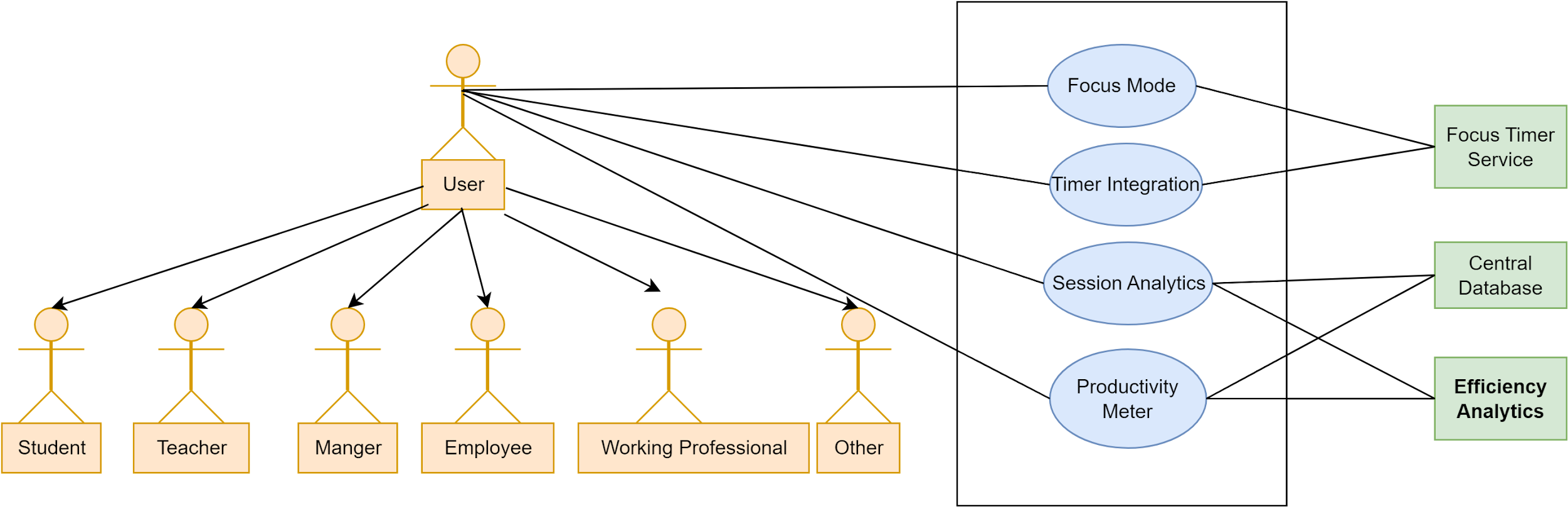
[](https://app.diagrams.net/?page-id=gi5pgJB8UzCgduBGHC_Y&scale=auto#G1Pxhi9rpScaylszASBdNGYxh-K-SoJM4E)

Figure 9: Deep Work Tracking Module

3.1.2.8 Level 1.8

**Name:**  Collaborative and Competitive Environment Creating Module

**PrimaryActor:** User(Student,Teacher,WorkingPrefessional,Employee,Managers,Others)

**Secondary Actor:** Central Database, Collaboration Engine

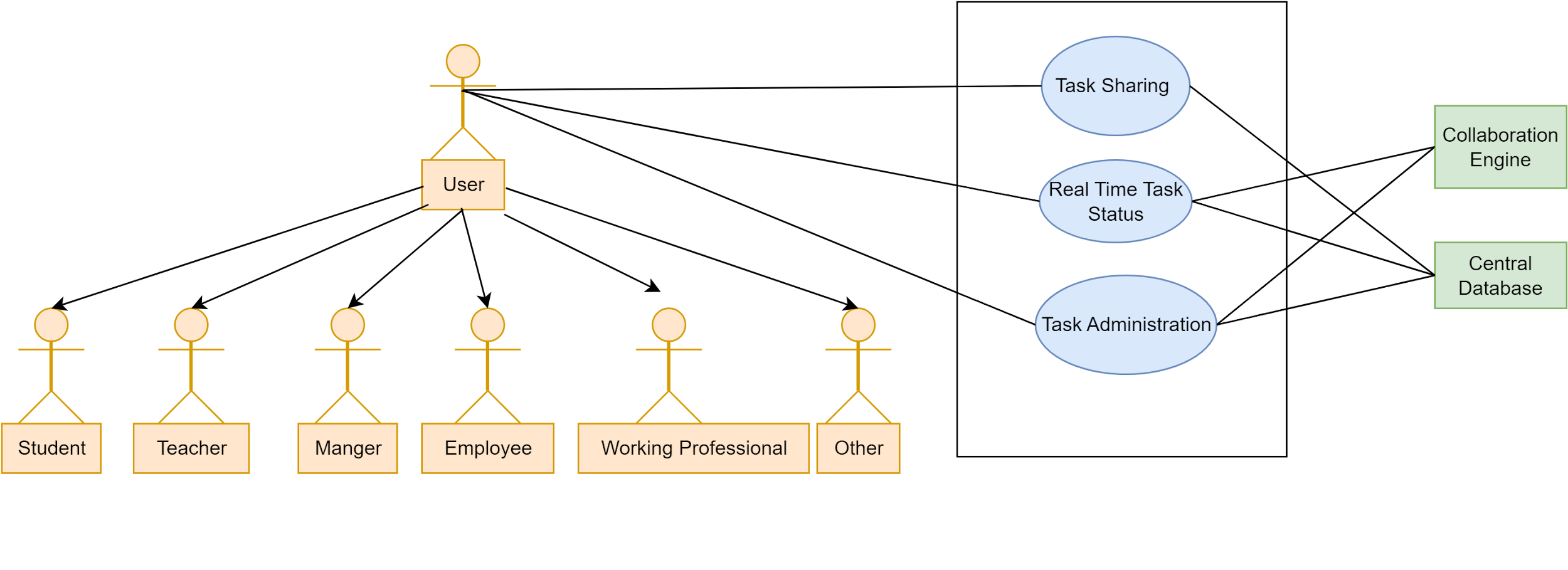
[](https://app.diagrams.net/?page-id=V-lxXwfAROI72Es8kEt4&scale=auto#G1h7G7C6jl2mQa-cB2pARjHSGyrcF-l4yk)

Figure 10: Collaborative and Competitive Environment Creating Module

4. Database Modeling

4.1 What is Database Modeling?

4.2 ThePurpose of Database Modeling

4.3 Database Modeling: TimeWise

4.3.1 Noun Listing

| No. | Nouns | Attributes  (Related Nouns) | Problem Space(P)/  Solution Space(S) |
| --- | --- | --- | --- |
| 01 | User | 2, 3, 4, 5, 6, 7, 24, 25 | P |
| 02 | Student | 1, 3, 4, 5, 6, 8, 13, 14, 21, 35, 36 | P |
| 03 | Teacher | 1, 2, 4, 5, 6, 8, 16, 32, 14, 15 | S |
| 04 | Manager | 1, 2, 3, 5, 6, 8, 14, 16, 31, 34 | P |
| 05 | Employee | 1, 2, 3, 4, 6, 8, 16, 31 | S |
| 06 | Working Professional | 1, 2, 3, 4, 5, 8, 10, 11, 12, 13 | S |
| 07 | Others | 1, 2, 3, 4, 5, 6, 8, 13, 16, 36 | S |
| 08 | Task | 2, 3, 4, 5, 6, 7, 21, 26, 27 | P |
| 09 | Progress | 2, 5, 8, 10, 37, 40, 52 | P |
| 10 | Performance | 2, 5, 6, 8, 14, 19, 37, 38 | P |
| 11 | Notification | 6, 12, 15, 23, 24 | P |
| 12 | Reminder | 6, 11, 29, 43 | P |
| 13 | Stats | 2, 8, 36, 37, 50 | S |
| 14 | Goal | 2, 3, 4, 8, 36 | S |
| 15 | Team | 1, 3, 4, 32, 47 | P |
| 16 | Feedback | 3, 5, 8, 38, 47 | S |
| 17 | Feedback Type | 16, 18 | S |
| 18 | Role | 1, 2, 3, 4, 5, 6 | S |
| 19 | Category | 2, 8, 37, 51 | S |
| 20 | Session | 1, 2, 3, 5, 45 | S |
| 21 | TimeWise | 1, 2, 3, 5, 8, 36 | S |
| 22 | SIgn In | 1, 23, 24 | P |
| 23 | Sign Up | 1, 24, 25 | P |
| 24 | Account | 1, 2, 3, 5, 23 | P |
| 25 | Admin | 1, 24, 32 | S |
| 26 | Deadline | 8, 27, 43 | S |
| 27 | Priority | 8, 26, 30 | S |
| 28 | Comments | 8, 32 | S |
| 29 | Timer | 12, 43 | S |
| 30 | Productivity Meter | 2, 10, 38, 44 | P |
| 31 | Task Sharing | 4, 8, 15, 47 | P |
| 32 | Task Administration | 3, 4, 8, 15, 25 | P |
| 33 | Analytics Engine | 13, 38, 41, 52 | S |
| 34 | User Dashboard | 1, 21 | S |
| 35 | Date and time service | 2, 8, 21, 26, 29 | S |
| 36 | Goal tracking | 2, 8, 14 | S |
| 37 | Progress Analytics | 9, 36, 44 | P |
| 38 | Performance Analytics | 10, 16, 37 | P |
| 39 | Reporting service | 33, 41, 52 | S |
| 40 | Performance logs | 10, 24 | P |
| 41 | Statistics engine | 13, 38, 39 | S |
| 42 | Visualization service | 9, 13, 33 | S |
| 43 | Focus timer service | 29, 30 | S |
| 44 | Efficiency analytics | 2, 10, 38 | S |
| 45 | Session Analytics | 20, 37 | S |
| 46 | Task status | 8, 31 | S |
| 47 | Collaboration engine | 1, 4, 15, 31 | S |
| 48 | Central database | 24, 40 | P |
| 49 | Task completion stats | 8, 13, 50 | S |
| 50 | Time spent stats | 8, 13, 44 | P |
| 51 | Category based stats | 19, 37 | S |
| 52 | Progress reports | 9, 33, 39 | S |
| 53 | System Services | 33, 42, 47 | S |

4.3.2 List of Data Objects

| **Data Objects** | **Attributes** |
| --- | --- |
| User  (Student,Teacher,Manager,  Employee,Working Professional,Other) | - User\_ID (Primary Key)  - Name  - Email  - Password  - Role  - Short Biodata  - User Status (e.g., Active, Inactive) |
| Task | - Task\_ID (Primary Key)  - Task Name  - Task Description  - Task Priority  - Task Deadline  - Task Owner (User\_ID)  - Task Participants (list of User\_IDs)  - Task Status (e.g., Pending, In Progress, Completed)  - Task Creation Date  - Task Modification Date  - Task Comments  - Task Category |
| Progress | - Progress\_ID (Primary Key)  - User\_ID (Foreign Key)  - Task\_ID (Foreign Key)  - Progress Amount (percentage or value)  - Progress History (log of changes)  - Progress Report (summary details) |
| Performance | - Performance\_ID (Primary Key)  - User\_ID (Foreign Key)  - Performance History (record of completed tasks and milestones)  - Performance Score (calculated metric)  - Performance Report (summary analysis) |
| Notification | - Notification\_ID (Primary Key)  - User\_ID (Foreign Key)  - Notification Type (e.g., Reminder, - Task Alert)  - Notification Message  - Notification Status (e.g., Sent, Pending, Read)  Timestamp |
| Reminder | - Reminder\_ID (Primary Key) -User\_ID (Foreign Key) -Task\_ID (Foreign Key) -Reminder Message -Reminder Date and Time -Reminder Frequency (e.g., Daily, Weekly) |
| Team | - Team\_ID (Primary Key)  - Team Name  - Team Members (list of User\_IDs)  - Created By (User\_ID)  - Creation Date  - Team Status (e.g., Active, Inactive) |
| Feedback | - Feedback\_ID (Primary Key)  - User\_ID (Foreign Key)  - Feedback Type (e.g., Task -Feedback, System Feedback)  - Feedback Message  - Date Submitted  - Rating (if applicable)  **-** Feedback Type(e.g., Task Feedback, System Feedback) |
| Goal | - Goal\_ID (Primary Key) - User\_ID (Foreign Key) - Goal Description - Goal Deadline - Progress Status (linked with progress - Goal Creation Date - Goal Completion Status - Goal Comments |
| Session | - Session\_ID (Primary Key)  - User\_ID (Foreign Key)  - Session Start Time  - Session End Time  - Duration  - Activity Summary |
| Focus Timer Service | - Timer\_ID (Primary Key)  - User\_ID (Foreign Key)  - Timer Start Time  - Timer End Time  - Break Intervals  - Session Notes  - Duration |
| Task Administration | - Task\_ID (Primary Key)  - Admin\_ID (Foreign Key)  - Admin Action (e.g., Modify, Delete)  - Action Timestamp  - Shared With (list of User\_IDs)  - Permissions (e.g., View, Edit)  - Sharing Date |
| Admin | - Admin\_ID (Primary Key)  - Name  - Email  - Password  - Admin Status (e.g., Active, Inactive) |
| Analytics Engine  (Session Analytics, Progress Analytics, Performance Analytics,Efficiency Analytics) | - Analytics Engine ID(Primary Key)  - User\_ID (Foreign Key)  - Analytics Engine Type(e.g., Session Analytics, Progress Analytics, Performance Analytics,Efficiency Analytics)  - Insight  - Generated Report  - Analysis Type (e.g., Efficiency, Time Spent)  - Data Source (e.g., Tasks, Activities) |
| System Services  (Date Time Service, Reporting Service, Visualization Service) | - Service\_ID(Primary Key)  - Analytics Engine ID(Foreign Key)  - Service Type(e.g.,Date Time Service, Reporting Service, Visualization Service)  - Data Source (e.g., Tasks, Activities) |
| Collaboration Engine | - Collaboration\_ID (Primary Key)  - User\_IDs (List of participants)  - Collaboration Type (e.g., Group Project, Task Collaboration)  - Creation Date  - Status (e.g., Active, Completed) |
| Statistics Engine | - StatEngine\_ID (Primary Key)  - User\_ID (Foreign Key)  - Stat Type (e.g., Task Completion, Time Spent)  - Calculation Date  - Stat Value |

### **4.3.3 Collections Schema**

**User**

{

"\_id": ObjectId,

"name": String,

"email": String,

"password": String,

"role": String,

"short\_biodata": String,

"user\_status": String

}

**Task**

{

"\_id": ObjectId,

"task\_name": String,

"task\_description": String,

"task\_priority": String,

"task\_deadline": Date,

"task\_owner": ObjectId,

"task\_participants": [ObjectId],

"task\_status": String,

"task\_creation\_date": Date,

"task\_modification\_date": Date,

"task\_comments": [String],

"task\_category": String

}

**Progress**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"task\_id": ObjectId,

"progress\_amount": Number,

"progress\_history": [String],

"progress\_report": String

}

**Performance**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"performance\_history": [String],

"performance\_score": Number,

"performance\_report": String

}

**Notification**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"notification\_type": String,

"notification\_message": String,

"notification\_status": String,

"timestamp": Date

}

**Reminder**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"task\_id": ObjectId,

"reminder\_message": String,

"reminder\_date\_time": Date,

"reminder\_frequency": String

}

**Team**

{

"\_id": ObjectId,

"team\_name": String,

"team\_members": [ObjectId],

"created\_by": ObjectId,

"creation\_date": Date,

"team\_status": String

}

**Feedback**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"feedback\_type": String,

"feedback\_message": String,

"date\_submitted": Date,

"rating": Number

}

**Goal**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"goal\_description": String,

"goal\_deadline": Date,

"progress\_status": String,

"goal\_creation\_date": Date,

"goal\_completion\_status": String,

"goal\_comments": [String]

}

**Session**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"session\_start\_time": Date,

"session\_end\_time": Date,

"duration": Number,

"activity\_summary": String

}

**Focus Timer Service**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"timer\_start\_time": Date,

"timer\_end\_time": Date,

"break\_intervals": Number,

"session\_notes": String,

"duration": Number

}

**Task Administration**

{

"\_id": ObjectId,

"task\_id": ObjectId,

"admin\_id": ObjectId,

"admin\_action": String,

"action\_timestamp": Date,

"shared\_with": [ObjectId],

"permissions": [String],

"sharing\_date": Date

}

**Admin**

{

"\_id": ObjectId,

"name": String,

"email": String,

"password": String,

"admin\_status": String

}

**Analytics Engine**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"analytics\_engine\_type": String,

"insight": String,

"generated\_report": String,

"analysis\_type": String,

"data\_source": String

}

**System Services**

{

"\_id": ObjectId,

"analytics\_engine\_id": ObjectId,

"service\_type": String,

"data\_source": String

}

**Collaboration Engine**

{

"\_id": ObjectId,

"user\_ids": [ObjectId],

"collaboration\_type": String,

"creation\_date": Date,

"status": String

}

**Statistics Engine**

{

"\_id": ObjectId,

"user\_id": ObjectId,

"stat\_type": String,

"calculation\_date": Date,

"stat\_value": Number

}

4.3.4 Collections Relationship Diagram

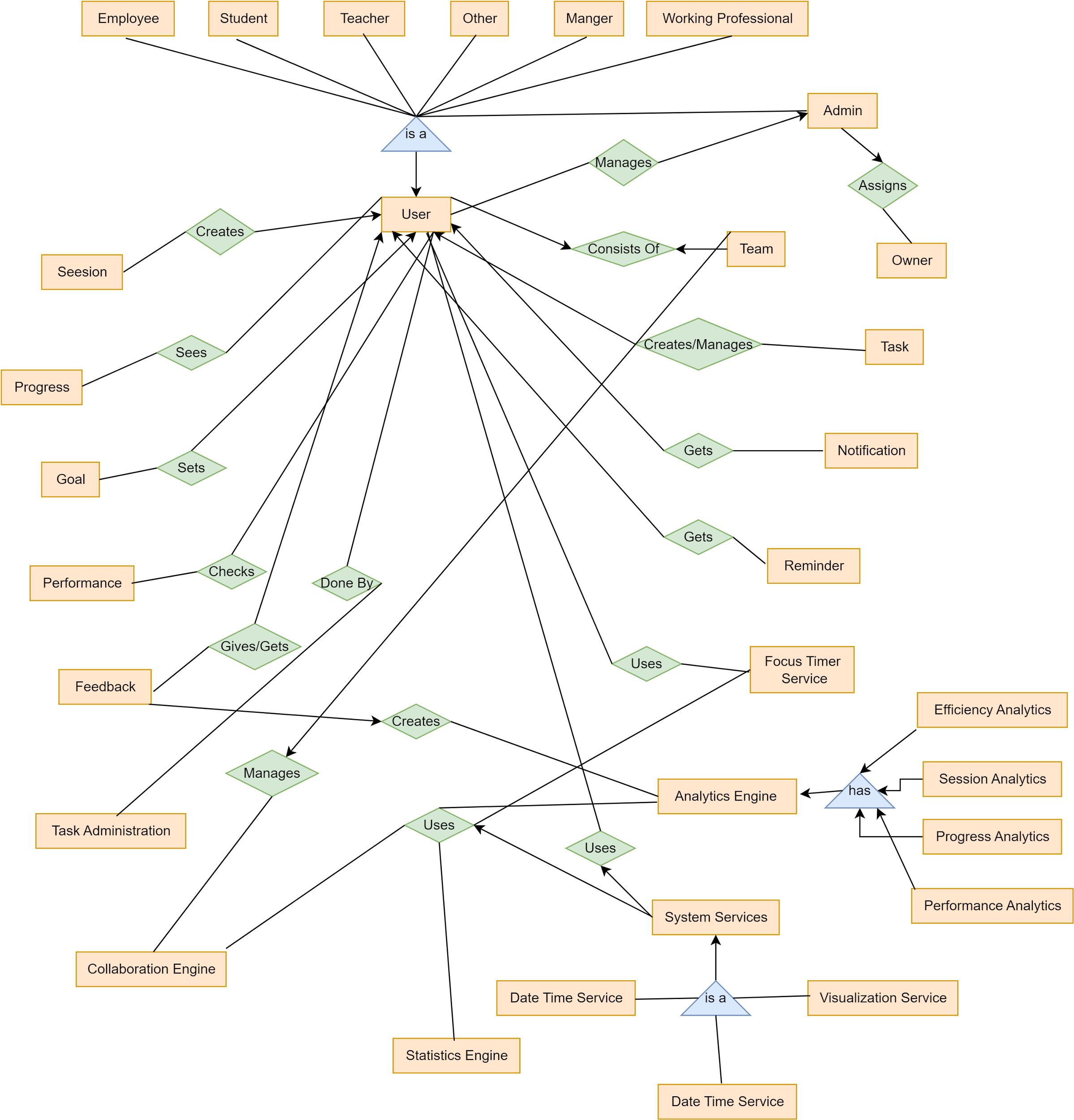
[](https://app.diagrams.net/?page-id=JpXTCvFZmwz3V7EVB828&scale=auto#G1e5l0_OCLXisfuB98qSUxReyMX_fn40B5)

Figure 11: TimeWise Collections Relationship Diagram

5. Class Based Modeling

5.1 What is Class Based Modeling?

5.2 Class Based Modeling: TimeWise

5.2.1 List of Nouns and Related Verbs

| No. | Nouns | Related Verbs |
| --- | --- | --- |
| 01 | User | Create, Manage, Update, Delete, Authenticate, Login, Logout, Register, View Profile, Interact, Collaborate |
| 02 | Student | Register, Manage Tasks, View Stats, Track Progress, Receive Notifications, Collaborate |
| 03 | Teacher | Register, Manage Tasks, View Stats, Assign, Collaborate, Provide Feedback |
| 04 | Manager | Assign, Track, Evaluate, Collaborate, Manage Tasks, Review, Provide Feedback |
| 05 | Employee | Register, Update, Manage Tasks, Track Progress, Provide Feedback |
| 06 | Working Professional | Manage Tasks, Track, View Stats, Collaborate, Receive Notifications |
| 07 | Others | Register, Manage Tasks, Track Progress, Collaborate, View Stats |
| 08 | Task | Create, Update, Delete, Assign, Track, Prioritize, Share, Complete, Modify, Comment |
| 09 | Progress | Track, Record, Evaluate, Update, Visualize |
| 10 | Performance | Evaluate, Track, Record, Analyze, Improve |
| 11 | Notification | Send, Receive, Track, Read, Acknowledge |
| 12 | Reminder | Set, Send, Track, Notify, Edit |
| 13 | Stats | Generate, View, Analyze, Interpret |
| 14 | Goal | Set, Track, Achieve, Edit, Evaluate, Complete |
| 15 | Team | Create, Manage, Update, Collaborate, Share |
| 16 | Feedback | Provide, Receive, Analyze, Track |
| 17 | Feedback Type | Select, Categorize |
| 18 | Role | Assign, Change, Manage |
| 19 | Category | Assign, Filter, Organize |
| 20 | Session | Start, End, Track, Analyze |
| 21 | TimeWise | Develop, Deploy, Manage, Use, Maintain |
| 22 | SIgn In | Authenticate, Validate |
| 23 | Sign Up | Register, Create Account |
| 24 | Account | Create, Update, Delete, Manage |
| 25 | Admin | Manage, Approve, Delete, Modify, Monitor |
| 26 | Deadline | Set, Track, Extend, Adjust |
| 27 | Priority | Set, Adjust, Change |
| 28 | Comments | Add, Edit, Delete, View |
| 29 | Timer | Start, Stop, Track, Adjust |
| 30 | Productivity Meter | Track, Display, Measure |
| 31 | Task Sharing | Share, Collaborate, Modify, View |
| 32 | Task Administration | Administer, Approve, Modify, Track, Delete |
| 33 | Analytics Engine | Analyze, Generate Reports, Track, Process |
| 34 | User Dashboard | View, Update, Manage |
| 35 | Date and time service | Track, Adjust, Provide |
| 36 | Goal tracking | Track, Analyze, Update |
| 37 | Progress Analytics | Analyze, Track, Interpret |
| 38 | Performance Analytics | Analyze, Track, Visualize |
| 39 | Reporting service | Generate, Deliver, Track |
| 40 | Performance logs | Record, Store, Track |
| 41 | Statistics engine | Process, Analyze, Generate |
| 42 | Visualization service | Render, Display, Update |
| 43 | Focus timer service | Start, Stop, Track, Adjust |
| 44 | Efficiency analytics | Analyze, Track, Improve |
| 45 | Session Analytics | Analyze, Track, Summarize |
| 46 | Task status | Update, View, Track |
| 47 | Collaboration engine | Facilitate, Manage, Coordinate |
| 48 | Central database | Store, Retrieve, Update, Manage |
| 49 | Task completion stats | Track, Record, Display |
| 50 | Time spent stats | Track, Analyze, Summarize |
| 51 | Category based stats | Generate, Track, Display |
| 52 | Progress reports | Generate, View, Track |
| 53 | System Services | Provide, Integrate, Manage |

5.2.2 General Classifications

Candidate classes are categorized based on the seven general classification. The analysis classes manifest themselves in one of the following ways:

1. External entities

2. Things

3. Events

4. Roles

5. Organizational units

6. Places

7. Structures

A candidate class is selected for special classification if it fulfills three or more characteristics

.

| **No.** | **Nouns** | **General Classification** |
| --- | --- | --- |
| 01 | User | 1, 4, 5 |
| 02 | Student | 1, 4, 5 |
| 03 | Teacher | 1, 4, 5 |
| 04 | Manager | 1, 4, 5 |
| 05 | Employee | 1, 4, 5 |
| 06 | Working Professional | 1, 4, 5 |
| 07 | Others | 1, 4 |
| 08 | Task | 2, 3, 4 |
| 09 | Progress | 3 |
| 10 | Performance | 3 |
| 11 | Notification | 3 |
| 12 | Reminder | 3 |
| 13 | Stats | 3 |
| 14 | Goal | 3 |
| 15 | Team | 1, 4, 5 |
| 16 | Feedback | 3, 5 |
| 17 | Feedback Type | 4 |
| 18 | Role | 4 |
| 19 | Category | 2 |
| 20 | Session | 3 |
| 21 | TimeWise | 1, 2, 5 |
| 22 | SIgn In | 3 |
| 23 | Sign Up | 3 |
| 24 | Account | 2 |
| 25 | Admin | 1, 4, 5 |
| 26 | Deadline | 3 |
| 27 | Priority | 3 |
| 28 | Comments | 3 |
| 29 | Timer | 3 |
| 30 | Productivity Meter | 3, 6 |
| 31 | Task Sharing | 2, 3 |
| 32 | Task Administration | 1, 4, 5 |
| 33 | Analytics Engine | 1, 2, 7 |
| 34 | User Dashboard | 2, 5, 7 |
| 35 | Date and time service | 2, 3, 5 |
| 36 | Goal tracking | 2, 3 |
| 37 | Progress Analytics | 2, 3, 5 |
| 38 | Performance Analytics | 2, 3, 5 |
| 39 | Reporting service | 2, 3, 5 |
| 40 | Performance logs | 3 |
| 41 | Statistics engine | 2, 5, 7 |
| 42 | Visualization service | 2, 5 |
| 43 | Focus timer service | 2, 3 |
| 44 | Efficiency analytics | 2, 3, 5 |
| 45 | Session Analytics | 2, 3, 5 |
| 46 | Task status | 2, 3 |
| 47 | Collaboration engine | 1, 2, 5 |
| 48 | Central database | 2, 5, 7 |
| 49 | Task completion stats | 3, 6 |
| 50 | Time spent stats | 3, 6 |
| 51 | Category based stats | 3, 5 |
| 52 | Progress reports | 2, 3, 5 |
| 53 | System Services | 1, 2, 5 |

5.2.3 Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria:

1. Retain information

2. Needed services

3. Multiple attributes

4. Common attributes

5. Common operations

6. Essential requirements

A candidate class generally becomes a class when it fulfills around three characteristics.

| No | Potential Classes | Selection Criteria |
| --- | --- | --- |
| 01 | User(Student,Teacher,Working Professional,Employee,Managers,Others) | 1, 3, 5, 6 |
| 02 | Task | 1, 2, 3, 5 |
| 03 | Progress | 1, 3, 5 |
| 04 | Performance | 1, 2, 3, 5, 6 |
| 05 | Reminder | 1, 3, 5 |
| 06 | Goal | 1, 2, 3, 5 |
| 07 | Team | 1, 2, 5, 6 |
| 08 | Feedback | 1, 3, 6 |
| 09 | Session | 1, 2, 5 |
| 10 | Account | 1, 5 |
| 11 | Analytics Engine  (Session Analytics, Progress Analytics, Performance Analytics,Efficiency Analytics) | 1, 2, 3, 5 |
| 12 | User Dashboard | 2 |
| 13 | System Services  (Date Time Service, Reporting Service, Visualization Service) | 2, 5, 6 |
| 14 | Progress Report | 1, 5 |
| 15 | Statistics Engine | 1, 2, 3, 5 |
| 16 | Collaboration Engine | 1, 2, 3, 5 |
| 17 | Focus Timer Service | 1, 2, 6 |
| 18 | Performance Report | 1, 5 |
| 19 | Task Administration | 1, 2, 6 |
| 20 | Task Completion Stats | 4, 5 |
| 21 | Time Spent Stats | 4, 5 |
| 22 | Category Based Stats | 4, 5 |
| 23 | Performance Logs | 4, 5 |
| 24 | Comments | 4, 5 |
| 25 | Admin | 1, 3, 5, 6 |
| 26 | Productivity Meter | 4, 5 |
| 27 | Timer | 4, 5 |
| 28 | Notification | 1, 3, 5 |

5.2.4 Selected Classes

There are 17 main and central classes in this system:

| **Selected Classes** |
| --- |
| User  (Student,Teacher,Manager,  Employee,Working Professional,Other) |
| Task |
| Progress |
| Performance |
| Notification |
| Reminder |
| Team |
| Feedback |
| Goal |
| Session |
| Focus Timer Service |
| Task Administration |
| Admin |
| Analytics Engine  (Session Analytics, Progress Analytics, Performance Analytics,Efficiency Analytics) |
| System Services  (Date Time Service, Reporting Service, Visualization Service) |
| Collaboration Engine |
| Statistics Engine |

5.3 Class-Card

### **1. User**

* **Attributes**:
  + Name (String)
  + Email (String)
  + Phone Number (String)
  + Gender (String)
  + FingerPrintData (Structure)
  + Date of Birth (Structure)
  + Role (Student, Teacher, Manager, Employee, Working Professional, Other)
  + Short Biodata (String)
  + User Status (Active, Inactive)
* **Methods**:
  + **Register()**: Register individual information.
  + **SetPassword()**: Set a password for the registered account (Admin responsibility).
  + **Authenticate()**: Authenticate whenever logged in (Admin responsibility).
  + **updatePersonalInformation()**: Update personal data (Admin responsibility).
* **Responsibilities**:
  + Responsible for registering personal data, maintaining user status, and managing tasks.
* **Collaboration**:
  + Collaborates with all the other modules and classes of the system to manage tasks.

### **2. Task**

* **Attributes**:
  + Task\_ID (Primary Key)
  + Task Name (String)
  + Task Description (String)
  + Task Priority (String)
  + Task Deadline (Date)
  + Task Owner (User\_ID)
  + Task Participants (list of User\_IDs)
  + Task Status (String)
  + Task Creation Date (Date)
  + Task Modification Date (Date)
  + Task Comments (String)
  + Task Category (String)
* **Methods**:
  + **CreateTask()**: Create a new task.
  + **UpdateTask()**: Update task details.
  + **AssignTask()**: Assign tasks to a user.
  + **ChangeTaskStatus()**: Change the status of the task (e.g., Pending, In Progress, Completed).
  + **AddComments()**: Add comments to the task.
* **Responsibilities**:
  + Responsible for managing tasks, including creation, updates, assignment, and status changes.
* **Collaboration**:
  + Collaborates with **User** to assign tasks and manage progress.
  + Administered by **Admin** for any system-level changes or critical updates.

### 

### **3. Progress**

* **Attributes**:
  + Progress\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Task\_ID (Foreign Key)
  + Progress Amount (Float)
  + Progress History (List of logs)
  + Progress Report (String)
* **Methods**:
  + **TrackProgress()**: Track the user's progress on a task.
  + **UpdateProgress()**: Update progress details based on user input.
  + **GenerateProgressReport()**: Generate a progress summary report.
* **Responsibilities**:
  + Responsible for tracking and reporting the progress of assigned tasks.
* **Collaboration**:
  + Collaborates with **User** to update and track their progress on specific tasks.
  + Linked with **Task** for progress tracking on specific tasks.

### **4. Performance**

* **Attributes**:
  + Performance\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Performance History (List of records)
  + Performance Score (Float)
  + Performance Report (String)
* **Methods**:
  + **EvaluatePerformance()**: Evaluate the user’s performance over a defined period.
  + **GeneratePerformanceReport()**: Generate a performance analysis report.
* **Responsibilities**:
  + Responsible for tracking and evaluating the overall performance of users based on completed tasks.
* **Collaboration**:
  + Collaborates with **User** to evaluate their performance.
  + Uses **Task** data for performance scoring and analysis.

### **5. Notification**

* Attributes:
  + Notification\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Notification Type
  + Notification Message
  + Notification Status
  + Timestamp
* **Methods:**
  + CreateNotification()
  + UpdateStatus()
  + DeleteNotification()
* **Responsibilities:**
  + Responsible for managing notifications sent to users.
  + Responsible for updating and tracking the status of notifications.
* **Collaboration:**
  + Collaborates with User to send notifications and update their status.
  + Works with Admin to manage the delivery and status of notifications.

### **6. Reminder**

* **Attributes:**
  + Reminder\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Task\_ID (Foreign Key)
  + Reminder Message
  + Reminder Date and Time
  + Reminder Frequency
* **Methods:**
  + CreateReminder()
  + UpdateReminder()
  + DeleteReminder()
* **Responsibilities:**
  + Responsible for managing reminders for users.
  + Responsible for ensuring reminders are sent at the appropriate times.
* **Collaboration:**
  + Collaborates with User to set and manage task reminders.
  + Collaborates with Task data to associate reminders with tasks.

### **7. Team**

* **Attributes:**
  + Team\_ID (Primary Key)
  + Team Name
  + Team Members
  + Created By
  + Creation Date
  + Team Status
* **Methods:**
  + CreateTeam()
  + UpdateTeam()
  + DeleteTeam()
* **Responsibilities:**
  + Responsible for managing team creation, member assignments, and status updates.
  + Responsible for ensuring teams are active and their members are tracked.
* **Collaboration:**
  + Collaborates with User to assign team members and manage team details.
  + Collaborates with Admin for team-related administrative tasks.

### 

### 

### **8. Feedback**

* **Attributes:**
  + Feedback\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Feedback Type
  + Feedback Message
  + Date Submitted
  + Rating
* **Methods:**
  + SubmitFeedback()
  + UpdateFeedback()
  + DeleteFeedback()
* **Responsibilities:**
  + Responsible for collecting and managing user feedback.
  + Responsible for ensuring feedback is properly tracked and stored.
* **Collaboration:**
  + Collaborates with **User** to collect feedback and manage its submission.
  + Collaborates with **Analytics Engine** to give personalized feedback.

### **9. Goal**

* **Attributes**:
  + Goal\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Goal Description
  + Goal Deadline
  + Progress Status
  + Goal Creation Date
  + Goal Completion Status
  + Goal Comments
* **Methods**:
  + CreateGoal()
  + UpdateGoal()
  + TrackProgress()
  + CompleteGoal()
* **Responsibilities**:
  + Responsible for managing user goals and tracking their progress.
  + Responsible for ensuring goals are completed on time and managing comments.
* **Collaboration**:
  + Collaborates with **Users** to set and track goals.
  + Works with **Progress** data to update the progress status of goals.

### **10. Session**

* **Attributes**:
  + Session\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Session Start Time
  + Session End Time
  + Duration
  + Activity Summary
* **Methods**:
  + StartSession()
  + EndSession()
  + UpdateSession()
* **Responsibilities**:
  + Responsible for tracking user sessions, including start and end times, activity, and duration.
  + Responsible for summarizing the activity performed during the session.
* **Collaboration**:
  + Collaborates with **User** to log and track session details.
  + Works with **Focus Timer Service** to track the duration of focus sessions.

### **11. Focus Timer Service**

* **Attributes**:
  + Timer\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Timer Start Time
  + Timer End Time
  + Break Intervals
  + Session Notes
  + Duration
* **Methods**:
  + StartTimer()
  + EndTimer()
  + PauseTimer()
  + ResetTimer()
* **Responsibilities**:
  + Responsible for managing and tracking the focus timer for users.
  + Responsible for ensuring that break intervals are maintained during the focus session.
* **Collaboration**:
  + Collaborates with **User** to start, pause, and end focus sessions.
  + Works with **Session** to log session time and track user progress.

### **12. Task Administration**

* **Attributes**:
  + Task\_ID (Primary Key)
  + Admin\_ID (Foreign Key)
  + Admin Action
  + Action Timestamp
  + Shared With
  + Permissions
  + Sharing Date
* **Methods**:
  + ModifyTask()
  + DeleteTask()
  + ShareTask()
* **Responsibilities**:
  + Responsible for managing and administering tasks.
  + Responsible for tracking modifications, deletions, and sharing of tasks.
* **Collaboration**:
  + Collaborates with **Team** to manage task actions and permissions.
  + Works with **User** to share tasks and assign appropriate permissions.

### **13. Admin**

* **Attributes**:
  + Admin\_ID (Primary Key)
  + Name
  + Email
  + Password
  + Admin Status (e.g., Active, Inactive)
* **Methods**:
  + CreateUser()
  + ModifyUser()
  + DeleteUser()
  + ManageTasks()
* **Responsibilities**:
  + Responsible for managing users and tasks, and maintaining administrative functions.
  + Responsible for setting and managing user permissions and roles.
* **Collaboration**:
  + Collaborates with **User** for user and task management..

### **14. Analytics Engine**

* **Attributes**:
  + Analytics Engine ID (Primary Key)
  + User\_ID (Foreign Key)
  + Analytics Engine Type (e.g., Session Analytics, Progress Analytics, Performance Analytics, Efficiency Analytics)
  + Insight
  + Generated Report
  + Analysis Type (e.g., Efficiency, Time Spent)
  + Data Source (e.g., Tasks, Activities)
* **Methods**:
  + GenerateReport()
  + AnalyzeData()
  + ProvideInsights()
* **Responsibilities**:
  + Responsible for analyzing user activities and generating insights and reports.
  + Responsible for providing detailed analytics based on user sessions, performance, and progress.
* **Collaboration**:
  + Collaborates with **User** for personalized analysis and feedback.
  + Works with **System Services** to visualize data and generate comprehensive reports.

### **15. System Services**

* **Attributes**:
  + Service\_ID (Primary Key)
  + Analytics Engine ID (Foreign Key)
  + Service Type (e.g., Date Time Service, Reporting Service, Visualization Service)
  + Data Source (e.g., Tasks, Activities)
* **Methods**:
  + ProvideDateTimeService()
  + GenerateReportService()
  + VisualizeData()
* **Responsibilities**:
  + Responsible for offering system services like date-time management, data reporting, and data visualization.
  + Responsible for supporting the Analytics Engine with service-based data processing.
* **Collaboration**:
  + Collaborates with **Analytics Engine** to process and visualize data for user analysis.

### **16. Collaboration Engine**

* **Attributes**:
  + Collaboration\_ID (Primary Key)
  + User\_IDs (List of participants)
  + Collaboration Type (e.g., Group Project, Task Collaboration)
  + Creation Date
  + Status (e.g., Active, Completed)
* **Methods**:
  + InitiateCollaboration()
  + ManageCollaboration()
  + TrackCollaborationProgress()
* **Responsibilities**:
  + Responsible for facilitating collaboration between multiple users for projects and tasks.
  + Manages and tracks the progress and status of collaborations.
* **Collaboration**:
  + Collaborates with **User** entities to involve participants in tasks or projects.
  + Works with **Task Administration** to oversee tasks assigned to collaborative groups.

### **17. Statistics Engine**

* **Attributes**:
  + StatEngine\_ID (Primary Key)
  + User\_ID (Foreign Key)
  + Stat Type (e.g., Task Completion, Time Spent)
  + Calculation Date
  + Stat Value
* **Methods**:
  + CalculateStatistics()
  + GenerateStatReport()
* **Responsibilities**:
  + Responsible for collecting, calculating, and analyzing statistics related to user activities.
  + Generates reports and provides statistical insights.
* **Collaboration**:
  + Collaborates with **User** for personalized statistical reports and analysis.
  + Works with **Analytics Engine** to provide statistical data for deeper insights and performance evaluation.

5.4 Class Responsibility(CRC) Diagram

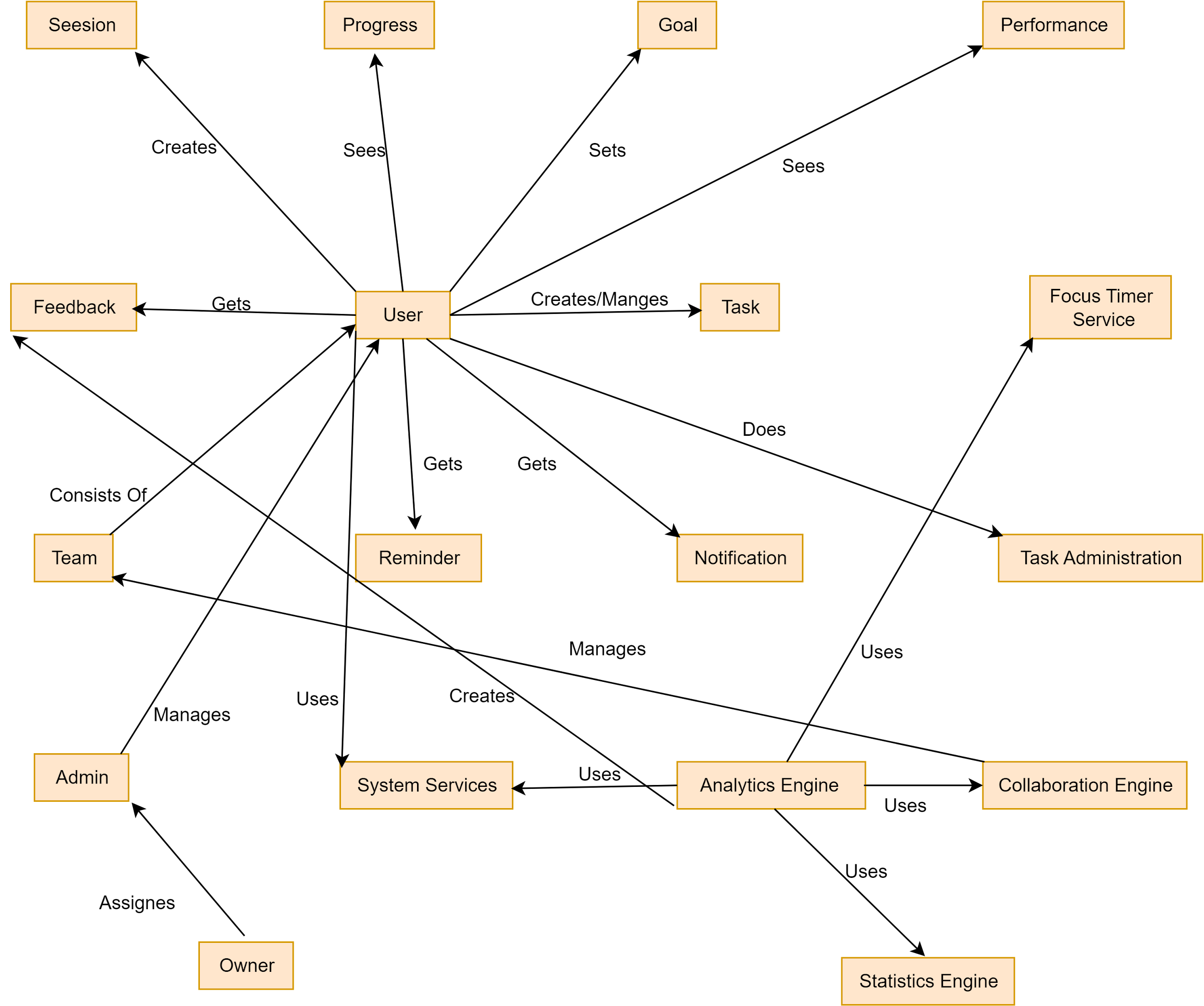
[](https://app.diagrams.net/?page-id=3EuqdE2Nsrsw9YRPW36l&scale=auto#G1RbcvLhmWMsykcjEAlYDjA8eqpI8_-lL2)

Figure 12: TimeWise CRC Diagram