Software Requirement Specification

TaskX - “Get organized. Work smarter. Stay Motivated”

Table of Contents

1. Introduction 4

1.1 Purpose

1.2 Project Scope

1.3 Intended Audience

2. Overall Description 6

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

3. Requirement Traceability Matrix 9

4. Functional Requirements 10

5. Non-functional Requirements 12

6. Use Case Diagram 15

7. ER Diagram - Logical 16

8. DataFlow Diagram 16

8.1 DataFlow Diagram Level - 0

8.2 DataFlow Diagram Level - 1

8.3 DataFlow Diagram Level - 2

09. State Diagram 19

10. Sequence Diagram 20

1. Introduction:

This section provides a brief description of the purpose of this SRS document along with the conventions used in it. It also provides details about the intended audience and the scope of the project.

1.1. Purpose

The Software Requirements Specification (SRS) document provides a comprehensive idea behind the scheduling system. This SRS document will provide a deeper and clearer understanding of what can be expected from this freshly introduced scheduling system to be established. This ensures that an optimized and error-free product is established which can be delivered to the user. It can even be used in the development stages for future work. The main objective here is to present this SRS to the customer for its approval which can in return provide a foundation for the development of the earlier version of the project to the development team. This creates a clear understanding between the user community and the development team. The SRS document will expand as the different contents of the system are explored from time to time with constant and constructive interactions between the user community and the development team.

1.2. Project scope

The scheduling system, ‘TaskX’ is a web application that can help users to create and maintain schedules for certain tasks in a smart and effective manner. TaskX is a free to use web-application which can be accessed through any web browser like Firefox, Chrome, etc. The main functionality being ‘create schedules’, TaskX helps the user to create schedules for certain tasks which include all sorts of activities like homework, assignments, exams, deadlines, meetings, and so on depending on the profession of the individual. It also provides several functionalities which can provide motivation to the user to follow the schedule that an individual has created.

The objective of the scheduling system is to schedule the tasks of an individual in such a manner that he/she is not exhausted/tired/bored by the schedule created and to maintain the consistency of the work done (tasks completed) by an individual at the end of the day. This would prevent the user from avoiding the work to be done on the tasks that he/she has planned/scheduled. Satisfaction, improvement in the way of managing the tasks, relief from the stress of pending tasks are some of the advantages that the project team wishes to provide to the users through the system. The system should be user-friendly, easy to use, able to improve the efficiency of the work, and have high customer satisfaction.

1.3. Intended audience

This project is intended to be used by a diverse group of people. This group includes Universities/schools where faculty, students can be the users, companies/industries where the staff, the manager can be the users, and many more such organizations or individuals. This SRS is intended for both the groups i.e. the user community which involves the customers, several individuals of the organizations as well as the project team involving the project manager, designer, developer, and tester.

u The customer will ensure that the product intended to be delivered is meaningful and acceptable or not through the SRS.

u The project manager will create plans, deadlines and keep track of the team individuals: whether he/she is contributing to the development of the project through the SRS.

u The designer will make use of the SRS as a foundation for the design of the system and will be in sync with the changes made in the SRS, ensuring that the customer's desires are fulfilled.

u The developer will make use of the SRS as a foundation for the development of functionalities of the system and will create a link between customer's requirements and the product features. SRS will help the developer to even maintain this link ensuring the customer's satisfaction.

u The tester will make use of the SRS to create test cases and scenarios as per the customer's request. When the desired functionality or feature is ready, the tester will check those cases and scenarios, ensuring that the product is fully functional and effective. This task is repeated as per the changes made in SRS. At the completion of the entire project, the tester again runs all the previously created tests to ensure that the customer's requirements are fulfilled.

Target Groups:

- Gender: Male/Female/Others
- Class: Lower/Middle/Upper Class
- Profession: Students/ Company Employees
- Geographic: Metro cities, University areas, etc.

2. Overall description:

2.1. Product perspective

The scheduling system is a smart and free-to-use web application that will be developed by the project team. It will provide the users an easy-to-use interface and attractive features. The users can easily get access to this web application via the internet. The system will be able to manage the scheduling of various tasks of a user and in turn, provide a smart schedule that will help the user with time management. The user will be able to manage the many tasks with the help of the schedule created through the system on a single platform which helps the user prevent unnecessary frustration or stress of creating a schedule for different tasks. Also, it helps the user to manage work properly and to get maximum work done with minimum exhaustion. This web application will be dependent on the data provided by the users, it will become necessary to store the data in a database. The web portal and the database will be connected via the internet for stable communication so that the data can be modified as per the situation.

2.2. Product features

The main feature of the scheduling system is to provide a smart and exhaustion-free schedule to the user based on the tasks entered by the user. The user for this product can be anyone: a student to manage the timetable, a professor to manage lectures, an employee to manage work, and so on. The minimalistic schedule provided by the system will be such that the user will not be exhausted by the timing of the activities nor will he/she end up wasting more time in the gaps between the activities. It will even provide YouTube links and podcasts to the user for staying focused and motivated for the tasks.

This product lets the registered users add tasks regarding different activities to get an optimal schedule which helps the user to finish the activities in time with minimal time wastage and also the schedule provided will be such that the user won't get tired or bored doing the tasks. These activities can be of different types like assignments, projects, exams, deadlines, and many more. The user can see the completed tasks, the pending tasks, and the entire schedule for all the tasks. The user can even modify a task at any time.

The web portal will provide the schedule, the upcoming tasks, and the remainder of the pending tasks as a notification through mail or even at the interface of the web application. It will also notify the user if there are any changes in the system.

2.3. User Classes & Characteristics

There is no bound to the type of user that can interact with the system. The user can be students, professors of a university or they can be the staff, manager of an office, even can be a pharmacist, doctor, and so on. As long as the activities are meaningful, the user can add the task related to the activity in the web application with necessary details like objective, deadline, etc. and an appropriate schedule will be presented by the website to the user.

A student of a university can add tasks regarding homework, assignments whereas the professor can add tasks regarding lectures, meetings, exams, tests, and many more.

A pharmacist can add tasks regarding the resupply of medicine stocks whereas a doctor, a patient can add tasks regarding appointments or medications and so on.

An employee at an office can add tasks regarding the deadline of some work whereas a manager can add tasks regarding the creation of new tasks to be achieved in the future.

It also doesn't have to be a superior-inferior relationship. Even an individual can add tasks regarding any upcoming event like a concert or a meetup with a friend or movie and so on. For any such appropriate and meaningful activities, the system will provide the best schedule to the user.

2.4. Operating environment

This is a web-based system and hence will require the operating environment for a client and server GUI.

- Software:

The website is designed to run on any platform which supports web-browsers like Google Chrome and Safari. A running mongoDB server if wanted to host it on localhost. No installation of the Database is required as mongoDB. MongoDB Atlas is the multi-cloud database service for MongoDB available on AWS, Google Cloud, and Azure. Best-in-class automation and built-in proven practices provide continuous availability, elastic scalability, and support with regulatory compliance.

- Hardware:

Operating System Supports all known operating systems, such as Windows, Linux, or Mac. Computer 512MB+ RAM, monitor with a minimum resolution of 1024x768, keyboard, and mouse.

2.5. Design and implementation constraints

This framework has a language limitation and is proposed to be executed in the 'English' language. Consequently, it doesn't offer multilingual help to clients. The Internet association is a requirement for the application. Since the application gets information from the data set over the Internet, it is significant that there is an Internet association for the application to work.

Choices with respect to which data set to utilize ought to be taken considering the way that information being traded or put away is enormous, and the suitable information the executives framework will yield effective execution.

2.6. Assumptions and dependencies

1. Assumptions:

Number	Description	Status	Reason for Assumption	Actions to validate	Impact
1	Active Internet Connection	Closed	The server can only be accessed while having an internet connection	Hosting the system on the server	The user is not able to access the system
2	User can move freely throughout the website	Closed	The user will be able to plan his day	Ability for the user to navigate	The user won't be able to plan his tasks
3	Availability of sufficient memory for the system	Closed	Non working of the system if the memory is not sufficient	Having sufficient storage space	System not working properly

2. Dependencies:

Number	Description	Status	Priority
1	System dependency on the Google Authentication for Login function	Closed	High
2	System Dependent on the internet based servers	Closed	Medium
3	Database dependent on the SQL servers	Closed	Medium

3. Requirement Traceability Matrix

REQUIREMENTS TRACEABILITY MATRIX

PROJECT TITLE **TaskX**

Business Requirements Documents (BRD)		Functional Specification Documents (FSD)		Test Case Documents (TCD)		
Business Requirement ID#	**Business Use Case**	**Functional Requirement ID#**	**Functional Requirement/ Use Case**	**Test Case ID#**	**Test Case**	**Status**
BR_001	User Interface by which the user and a computer system interact, in particular the use of input devices and software	FR_001	Responsiveness of the web application	TC_001	Web application run in laptop view	Implemented
				TC_002	Web application run in mobile view	Implemented
				TC_003	Web application run in tablet view	Implemented
BR_002	User can login/signup into the platform using Gmail-ID and password	FR_002	Login and Sign-up Module	TC_004	Login/Sign-up Security	Implemented
				TC_005	Field Validation- G-mail ID	Implemented
				TC_006	Field Validation- Password	Implemented
BR_003	Interaction modules by which use can interact with the system	FR_003	Add Tasks Module	TC_007	If new user, add task	Implemented
				TC_008	If old user, add task taking into consideration of already scheduled task	Implemented
		FR_004	Arrange Meeting with team-mates	TC_009	Checking free slots of all the attendees	Implemented
		FR_005	Edit Tasks Module	TC_010	One Field is changed	Implemented
				TC_011	More than one field is changed	Implemented
				TC_012	No fields changed	Implemented
				TC_013	Scheduled Tasks Deleted	Implemented
				TC_014	No Scheduled slots/ past history cannot be edited	Implemented
		FR_006	Selecting Users/ area of interest	TC_015	Response when selecting the tags	Implemented
BR_004	View only modules by which use can only view modules as a feedback from the system	FR_007	View Scheduled Tasks Module	TC_016	Checking slots of sessions divided from a particular task	Implemented
		FR_008	View Task Progress Module	TC_017	Analysing the progression of on going/completed tasks	Implemented
		FR_009	View Podcasts/Videos Module	TC_018	Making surity of Valid Links	Implemented
				TC_019	Links are fetched according to tags	Implemented

Original File Link (For Better Quality):
https://drive.google.com/file/d/1tAxl_2W8xQcMcTntQ_WAETvEqr0Bp4HD/view?usp=sharing

4. Functional Requirements

4.1. Functional Requirement 1.1

- ID: FR1
- Title : Responsiveness of the web-application
- Description: A user should be able to not only view an application but also correctly view the application in terms of the layout for different mediums such as mobile phone, laptop, desktop, tablet, etc.
- Rational: In order for a user to get a responsive view of the web-application
- Dependency: None

4.2 Functional Requirement 1.2

- ID: FR2
- Title: Log-In & Signup Module
- Description: A login is a set of credentials used to authenticate a user. These consist of a username and password of Google account.
- Rational: In order for a user to sign-in/log-in the web application.
- Dependency: FR1

4.3 Functional Requirement 1.3

- ID: FR3
- Title : Add tasks Module
- Description: After logging in the system, a user should be able to schedule a task according to his free time by inputting necessary information like Task Description, Task Name, Category, Duration, End Time & Priority of the task.
- Rational: In order for a user to to add a task into the system.
- Dependency: FR1, FR2

4.4 Functional Requirement 1.4

- ID: FR4

- Title : Arrange meeting with team-mates
- Description: A user should be able to arrange meetings with the help of the scheduler which yields the lists of free slots among team members.
- Rational: In order for a user to arrange a platform for meeting among team members.
- Dependency: FR1, FR2, FR3

4.5 Functional Requirement 1.5

- ID: FR5
- Title: Edit tasks module
- Description: A user once logged in and has already scheduled a task, should be able to edit the scheduled task on the basis of one field or more than one field (fields refers to the information added while adding a task).
- Rational: In order for a user to edit an already scheduled task.
- Dependency: FR1, FR2, FR3, FR4

4.6 Functional Requirement 1.6

- ID: FR6
- Title: Selecting User's area of interests
- Description: User should be able to select the tags after 1st sign-up, in order to get a list of recommended podcasts and helpful videos.
- Rational: In order for a user to get tags as an area of interests.
- Dependency: FR1, FR2, FR3

4.7 Functional Requirement 1.7

- ID: FR7
- Title : View Scheduled Tasks module
- Description: A user once logged in and has already scheduled a task, should be able to view his/her scheduled tasks based on TaskID.
- Rational: In order for a user to view the scheduled tasks.
- Dependency: FR1, FR2, FR3

4.8 Functional Requirement 1.8

- ID: FR8

- Title: View Task Progress Module
- Description: A user once logged in and has already scheduled a task, User should be able to get a full list of progression of current tasks, if he/she misses any of the tasks, the user can re-schedule the task again.
- Rational: In order for a user to get an idea about progression of tasks.
- Dependency: FR1, FR2, FR7

4.9 Functional Requirement 1.9

- ID: FR9
- Title : View Podcasts/ Video Module
- Description: A user once logged in and has already selected the area of interest, user should be able to get a list of podcasts and helpful videos.
- Rational: In order for a user to view podcasts/videos according to the area of interests entered.
- Dependency: FR1, FR2, FR6

5. Non-Functional Requirements

5.1 Performance:

- ID: NR1
- TAG: Response Time
- Description: The response time of the system while moving between two or more web-pages.
- Ideal: Transition between web-pages should take place in not more than 2 seconds.
- Valid Scenario: Transition between web-pages should take place in not more than 5 seconds.

5.2 Security:

- ID: NR2
- TAG: User Login
- Description: If a user tries to login with some other email ID not registered with the system, an error message should be displayed notifying the user regarding it and the user should not be allowed to log in.

- Ideal & Valid Scenario: A user should not be allowed to log in all of the cases.

5.3 Maintainability:

- ID: NR3
- TITLE: Application Extensibility
- Description: The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.
- Ideal & Valid Scenario: Future modules should be easily adapted to the system and dependencies across the already existing system modules should be as low as possible.

5.4 Testability:

- ID: NR4
- TITLE: Application Testability
- Description: Software testability is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context.
- Ideal & Valid Scenario: Testability of the software artifact should be as high as possible so that finding faults in the software (if it has), by means of testing is easier.

5.5 Availability:

- ID: NR5
- TITLE: System Availability
- Description: Software Availability is defined as the probability that the system is operating properly when it is requested for use (network failure not under consideration).
- Ideal & Valid Scenario: 100% of the time, a system should be available.

5.6 Dependability:

- ID: NR6
- TITLE: System Dependability
- Description: Software dependability is the ability to provide services that can defensibly be trusted within a time-period. A user should be notified of strangeness that happen
- Ideal Scenario: 100% of the time, a system should be dependable .

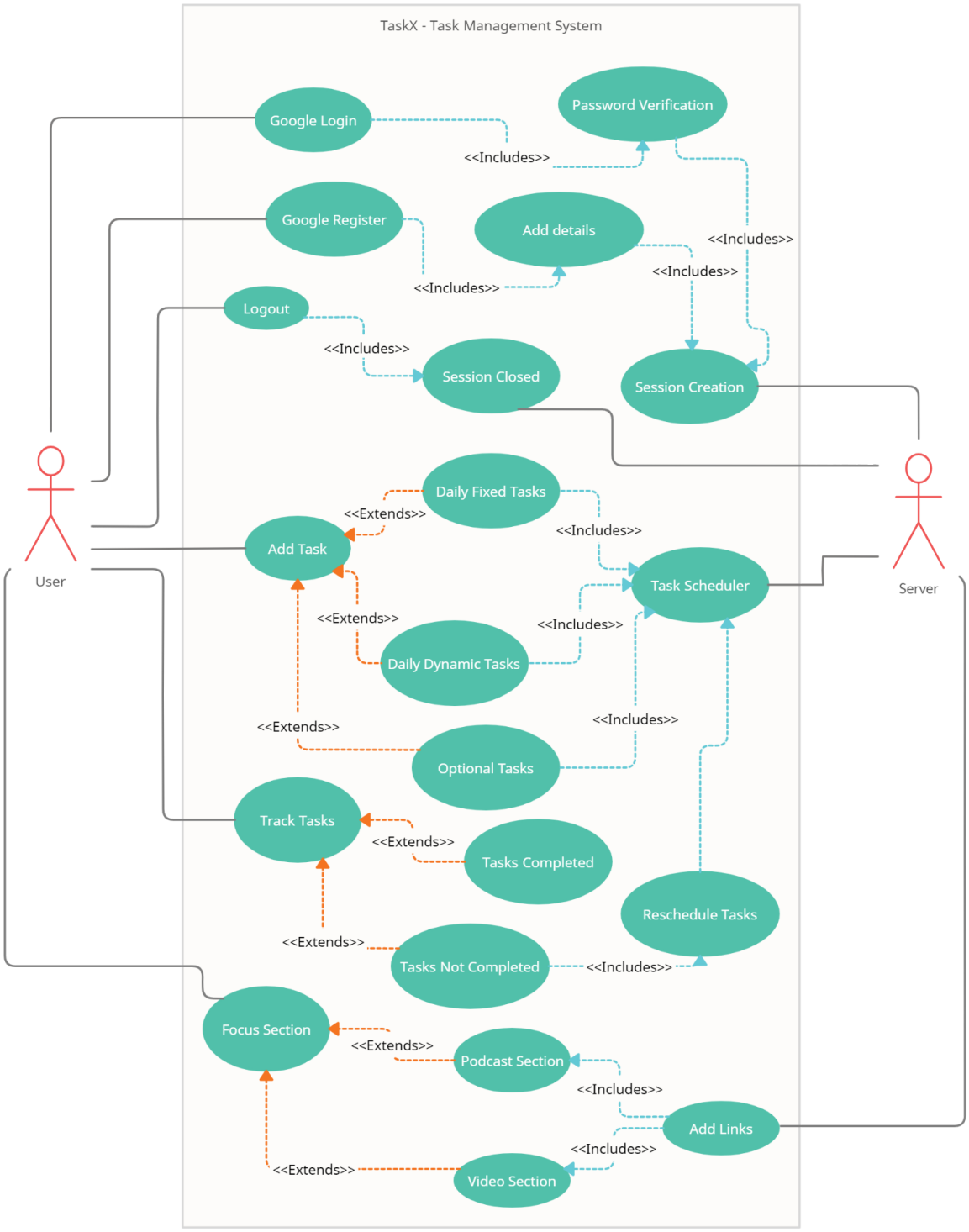
5.7 Compatibility:

- ID: NR7
- TITLE: System Compatibility
- Description: Software Compatibility is to determine whether the software application is proficient enough to run in different browsers, database, hardware, operating system, mobile devices, and network
- Ideal & Valid Scenario: 100% of the time, a system should be compatible.

5.8 Usability:

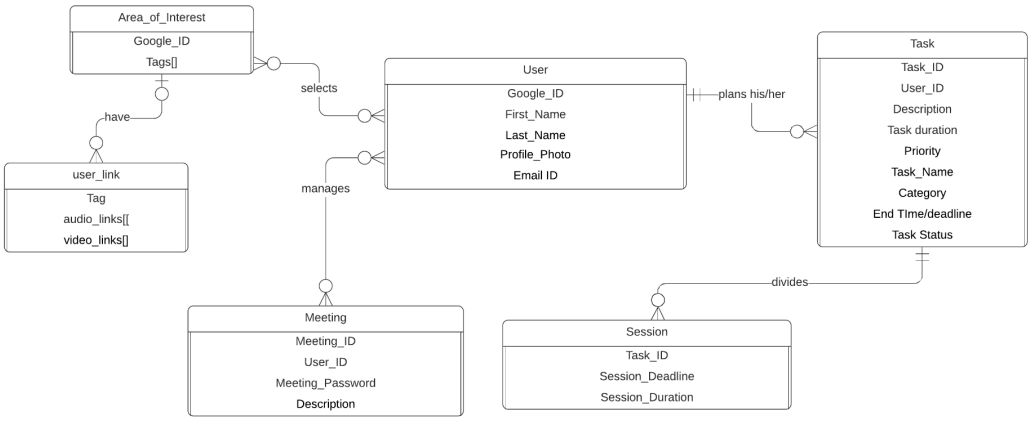
- ID: NR8
- TITLE: System Usability
- Description: Software Usability can be described as the capacity of a system to provide a condition for its users to perform the tasks safely, effectively, and efficiently.
- Ideal & Valid Scenario: A system should be usable for 100% of the time.

6. Use Case



For a better view:
https://drive.google.com/file/d/1Jv4fSVzm16QAjzI4LOeTPRViMxsO1_HC/view?usp=sharing

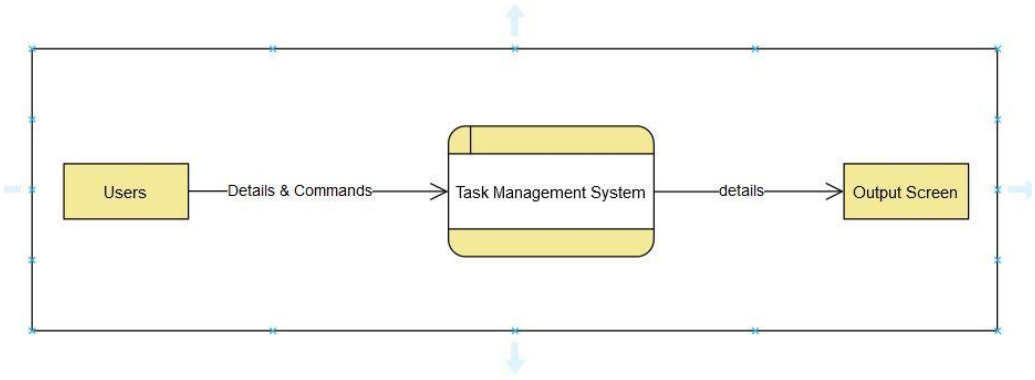
7. Logical ER Diagram:



For a better view:
https://drive.google.com/file/d/1sj_zqI9B3rSe2etS4ZkCs3EDlaJYdQJk/view?usp=sharing

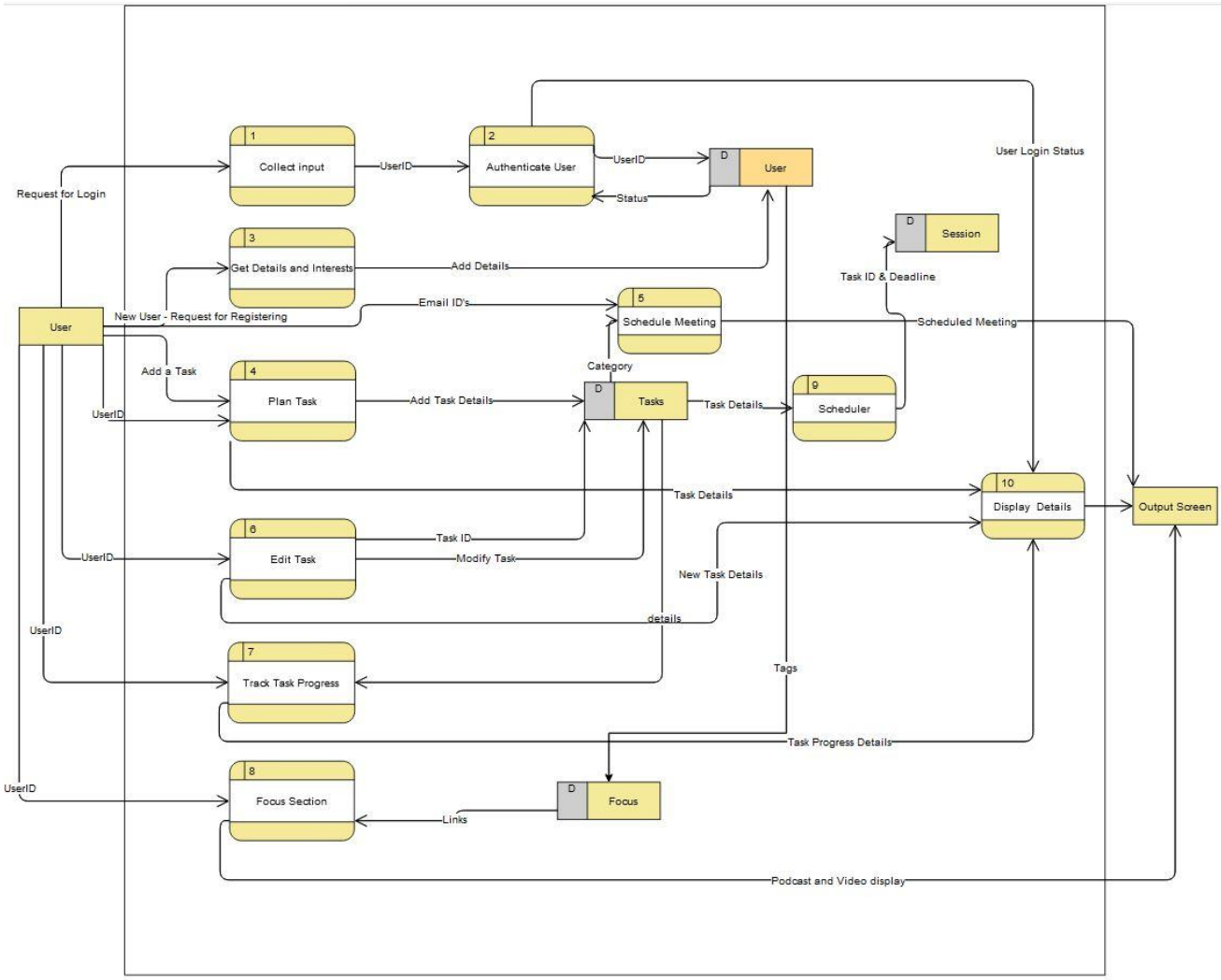
8. Data Flow Diagram:

8.1 DFD Level - 0:



For a better view:
https://github.com/TaskX-SE/TaskX/blob/master/DFD_Level-0.jpeg

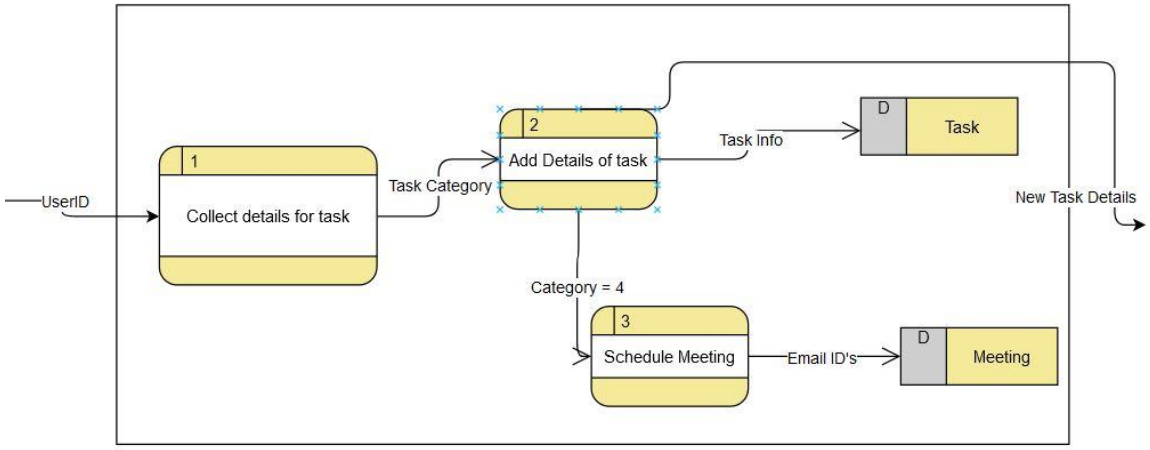
8.2 DFD Level - 1:



For a better view:
https://github.com/TaskX-SE/TaskX/blob/master/DFD_Level-1.jpeg

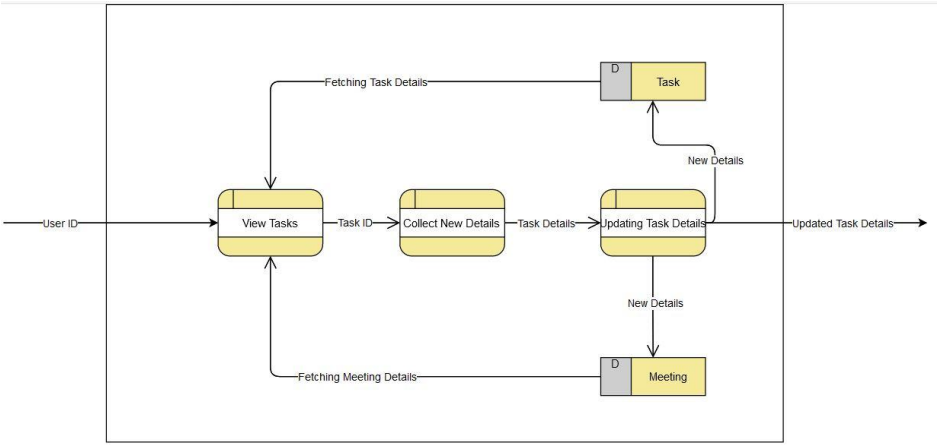
8.3 DFD Level - 2:

8.3.1 DFD Level - 2: Add task



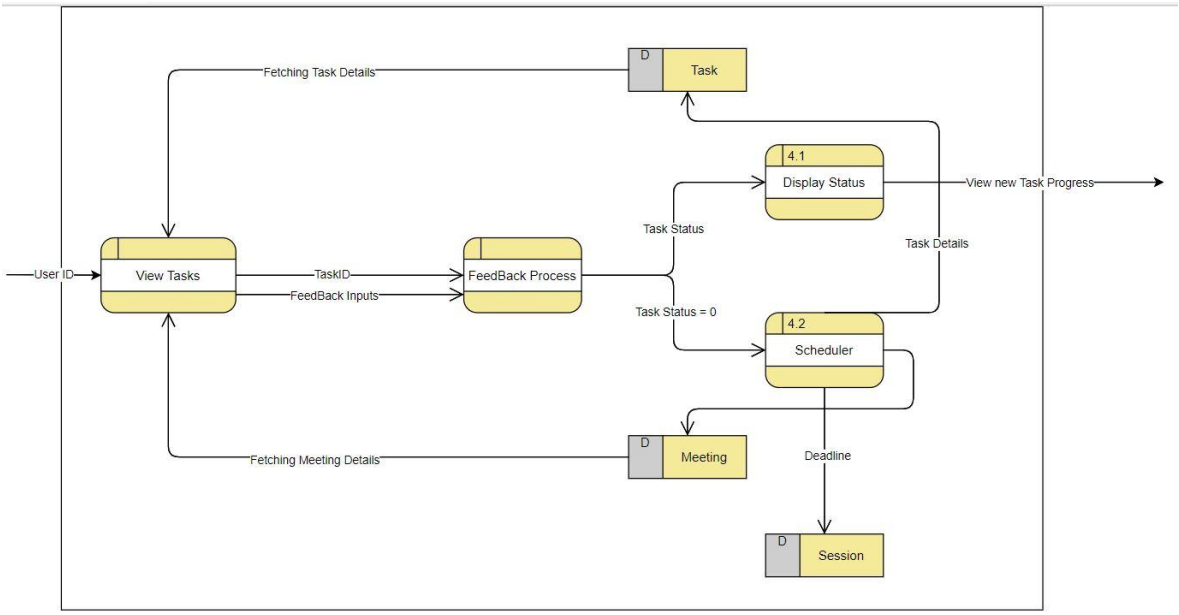
For a better view:
https://github.com/TaskX-SE/TaskX/blob/master/DFD_Level-2_PlanTask.jpeg

8.3.2 DFD Level - 2: Edit task



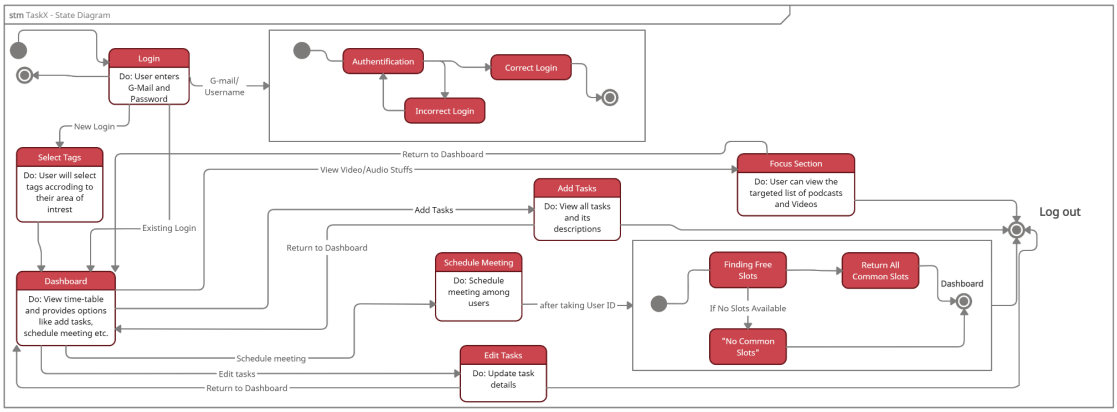
For a better view:
https://github.com/TaskX-SE/TaskX/blob/master/DFD_Level-2_EditTask.jpeg

8.3.3 DFD Level - 2: Track task progress



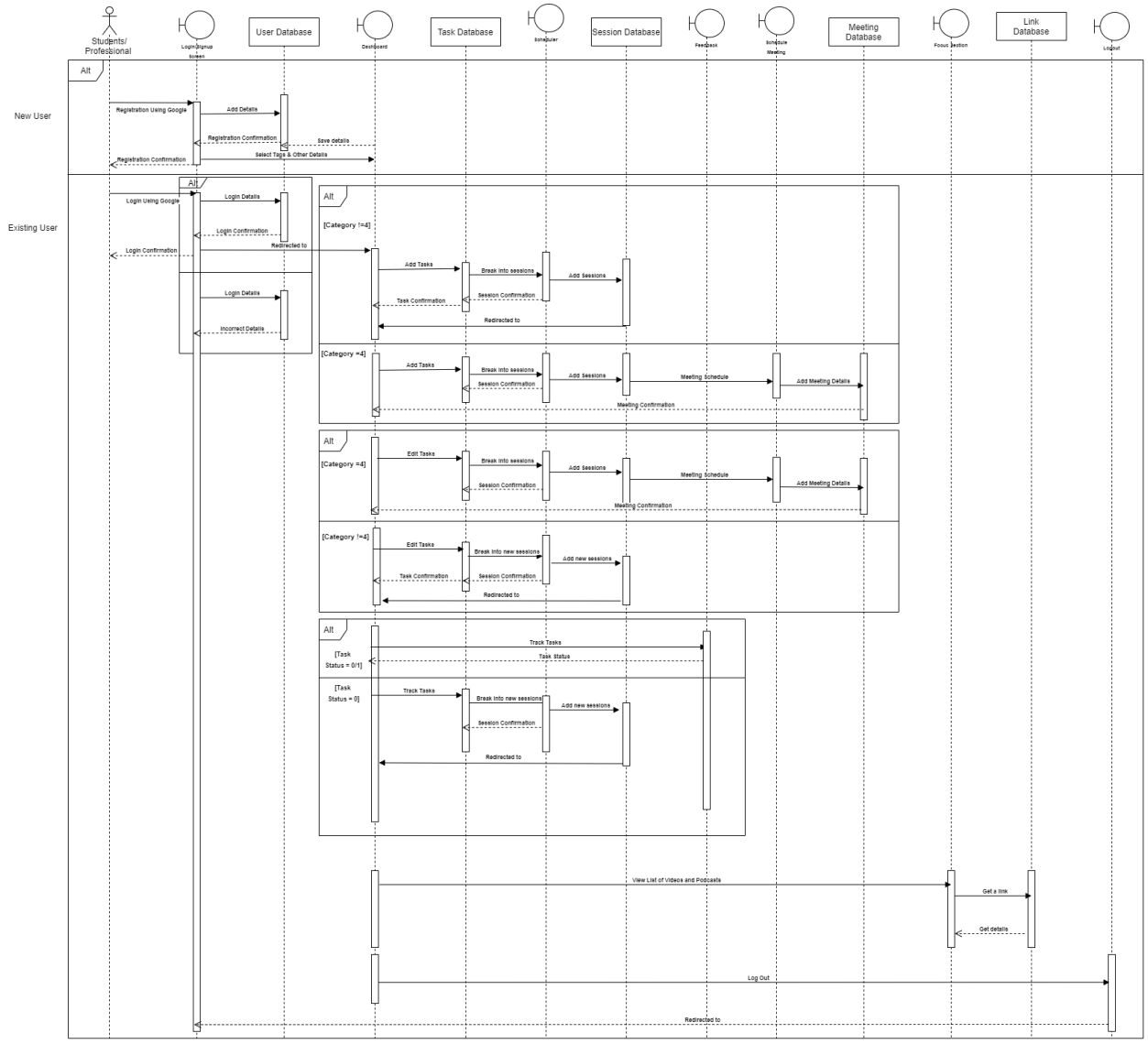
For a better view:
https://github.com/TaskX-SE/TaskX/blob/master/DFD_Level-2_TrackTask.jpeg

9. State Diagram:



For a better view:
https://drive.google.com/file/d/13Nejhsz1LSaiJOnGQ9bj01RXeDRC4lsx/view?usp=sharing

10. Sequence Diagram:



Link For a better view:
https://drive.google.com/file/d/1bGHecKKeFhA_eV4maYJTSZpkp_-8zdMv/view?usp=sharing