Task Manager

Patrick Jason J

MEPCO Schlenk Engineering College

jpatrickjason@gmail.com

Task Manager

Architecture/Design Document

Table of Contents

1 Introduction:

Task Manager is a web application designed using Django which assists the people to create, manage tasks and teams by adding users, assigning the tasks and monitoring the status of the tasks assigned to them.

This document describes the architecture and design for the Task Manager application being developed for FOSEE IITM , to create, manage tasks and teams.

The purpose of this document is to describe the architecture and design of the Task Manager application in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

- Users and the customer – they want assurances that the architecture will provide for system functionality and exhibit desirable non-functional quality requirements such as usability, reliability, etc.
- Developers – they want an architecture that will minimize complexity and development effort.
- Project Manager – the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies. For this to happen, the modules need well-defined interfaces. Also, because most individuals specialize in a particular skill or technology, modules should be designed around specific expertise. For example, all UI logic might be encapsulated in one module. Another might have all business logic.
- Maintenance Programmers – they want assurance that the system will be easy to evolve and maintain on into the future.

The architecture and design for a software system is complex and individual stakeholders often have specialized interests. There is no one diagram or model that can easily express a system's architecture and design. For this reason, software architecture and design is often presented in terms of multiple views or perspectives [IEEE Std. 1471]. Here the architecture of the Task Manager application is described from 4 different perspectives [1995 Krutchen]:

1. Logical View – major components, their attributes and operations. This view also includes relationships between components and their interactions. When doing OO design, class diagrams and sequence diagrams are often used to express the logical view.

2. Process View – the threads of control and processes used to execute the operations identified in the logical view.
3. Development View – how system modules map to development organization.
4. Use Case View – the use case view is used to both motivate and validate design activity. At the start of design the requirements define the functional objectives for the design. Use cases are also used to validate suggested designs. It should be possible to walk through a use case scenario and follow the interaction between high-level components. The components should have all the necessary behavior to conceptually execute a use case.

2 Design Goals

There is no absolute measure for distinguishing between good and bad design. The value of a design depends on stakeholder priorities. For example, depending on the circumstances, an efficient design might be better than a maintainable one, or vise versa. Therefore, before presenting a design it is good practice to state the design priorities. The design that is offered will be judged according to how well it satisfies the stated priorities.

The design priorities for the Task Manager application are:

- The design should minimize complexity and development effort.
- The design should be reusable and this app should be fit it any other Django sites.

This application is designed in a such a way it is reusable so it is built as three Django apps

'tasks', 'accounts' and 'teams'.

3 System Behavior

The use case view is used to both drive the design phase and validate the output of the design phase. The architecture description presented here starts with a review of the expect system behavior in order to set the stage for the architecture description that follows. For a more detailed account of software requirements, see the requirements document.

The users and allowed to create teams by adding members to their team and assign them specific tasks and also could monitor the status of the tasks.

4 Logical View

The logical view describes the main functional components of the system. This includes modules, the static relationships between modules, and their dynamic patterns of interaction.

In this section the modules of the system are first expressed in terms of high level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

4.1 High-Level Design (Architecture)

The high-level view or architecture consists of 3 major components:

Fig 1 System behavior.

- The **User** is the end user who uses the Task Manager application.
- The **Application control logic** is the main driver of the application. It presents information to the users and reacts to user inputs.
- The **Database** is a central repository for data of the tasks and teams.

4.2 Mid-Level Design

There are three Django apps with models for each one.

- Accounts – Users
- Tasks -Tasks, Comments
- Teams - Teams

The views in the apps:

Accounts :

- SignUp
- Sign-in(built-in Django authentication).

Tasks:

Class Based Views:

- TasksListView
- TaskDetailView
- CreateTasksView
- TasksUpdateView

- TasksDeleteView

Function Based Views:

- add_comment_to_task
- comment_remove

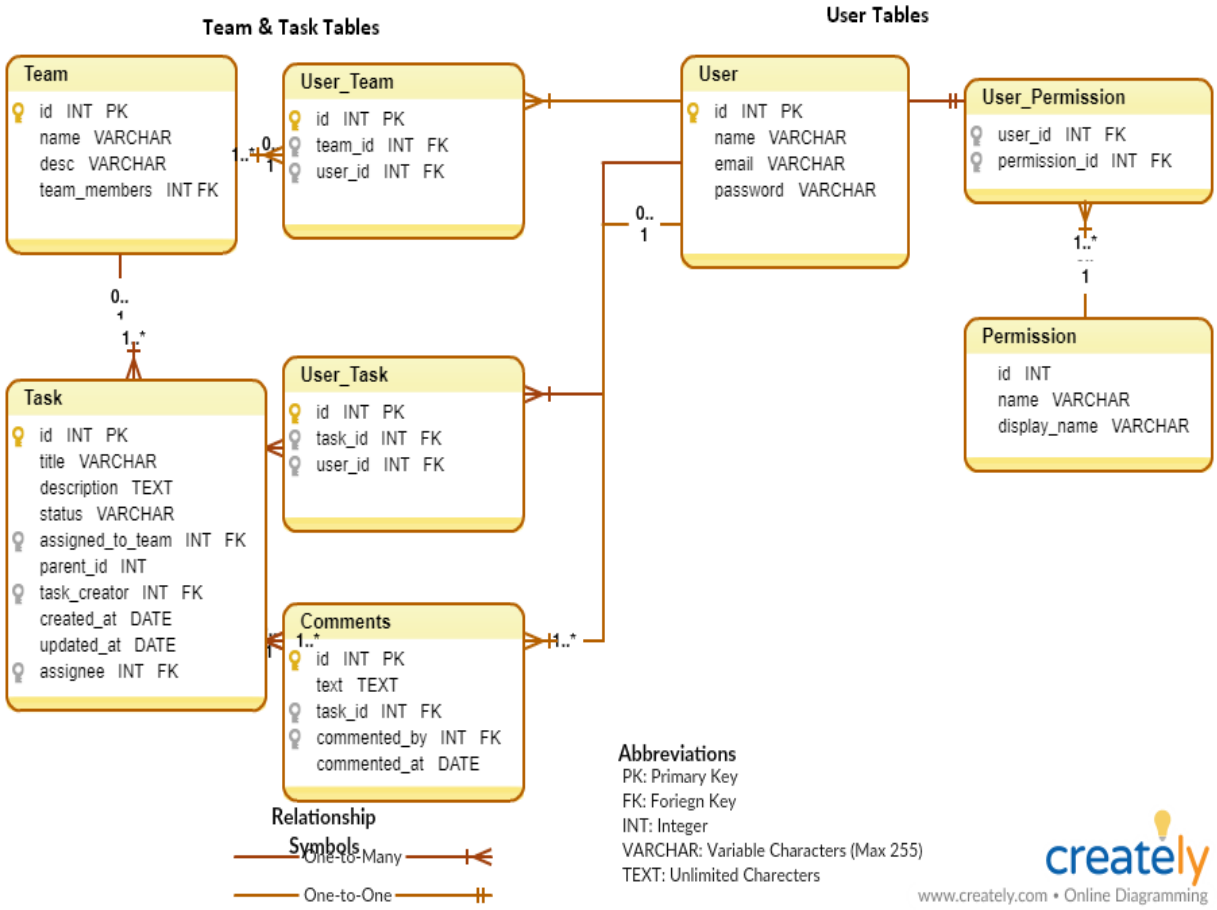
Teams:

Class Based Views:

- CreateTeams.
- SingleTeams.
- TeamsDeleteView

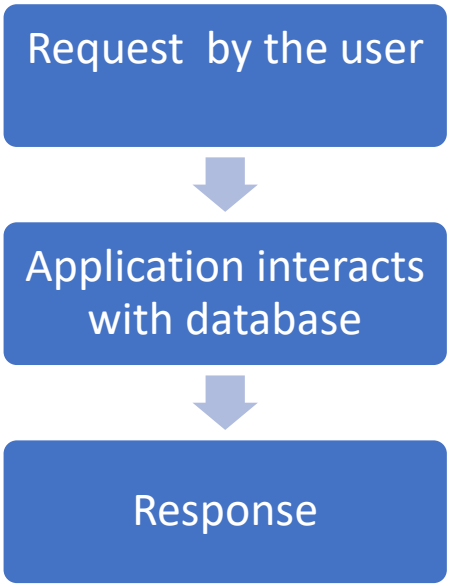
4.3 Detailed Class Design

The UML diagram of the database is given below



5 Process View

The process flow of the application is given below



6 Physical View

[TBD]

7 Use Case View

The functionality available in the apps are listed below:

Teams:

- Create Teams
- Delete Teams

Tasks:

- Create tasks
- Delete tasks
- Edit tasks
- Add assignee to tasks
- Comment on tasks

Accounts:

- Login
- Signup