Domain Analysis Model

The Domain Analysis Model is a systematic approach for analyzing and understanding the key concepts, relationships, and constraints within a particular problem domain, often used in software engineering.

1. Introduction

A web-based project management tool's domain analysis model attempts to give users a thorough understanding of the small company project management domain. In order to design a user-friendly and efficient project management solution, the model explains the main ideas, entities, interactions, and procedures involved in project management for small firms.

2. General Information About Domain

Planning, organizing, and coordinating resources to accomplish particular goals and objectives within a small business environment is referred to as small business project management. In order to complete a project and get the desired result, this area entails managing projects with constrained resources, such as time, money, and staff. Establishing project scope, developing project plans, allocating resources, controlling schedules, and tracking development are common components of small business project management. In small enterprises, efficient project management can boost output, revenue, and client happiness. Due to scarce resources, a lack of specialised knowledge, and conflicting priorities, it can be difficult as well.

3. User

- **Project manager**: The project manager, who is responsible for overseeing the project and ensuring that it is completed on time and within budget.
- **Team members**: The tool will also be used by the various team members who are working on the project, including designers, developers, writers, marketers, and others.
- **Business owners**: Business owners who need to manage multiple projects at once may also find a web-based project management tool to be useful.

4. Existing Software

There are several software solutions available in the market for web-based project management for small businesses. Some of the most popular ones include Trello, Asana, Basecamp, Wrike, and Monday.com.

Boundary, Entity and Country Object

1. Boundary Object

A boundary object is an object that acts as a link or middleman between various communities or groups of people with various viewpoints, values, or behaviours. It is a shared resource that makes cross-border communication and cooperation possible. A boundary object in the context of a project could be any real or intangible asset that helps project stakeholders with various backgrounds and interests work together and understand one another, including a document, tool, model, or other boundary items.
Web Browser: The user interacts with the system through a web browser.
Database: The system might access the database to provide information to the user.

2. Entity Object

Conversely, an entity object is an abstraction of a real-world item or notion with properties or attributes. An entity object represents a data entity with a distinct identifier and a collection of attributes that specify its properties or behaviours in software engineering or database architecture. An entity object in the context of a project could be any tangible or ethereal object pertinent to the project's objectives and scope.

- **Projects**: The project entity is an initiative that needs to be planned, executed and monitored
- **Task**: The task entity refers to a specific task that needs to be done in a specific time period.
- **Resource** : The resource entity refers to resources that are required to complete the project.
- **Report**: It refers to the information from the project management tool to identify issues,monitor the progress and make informed decisions

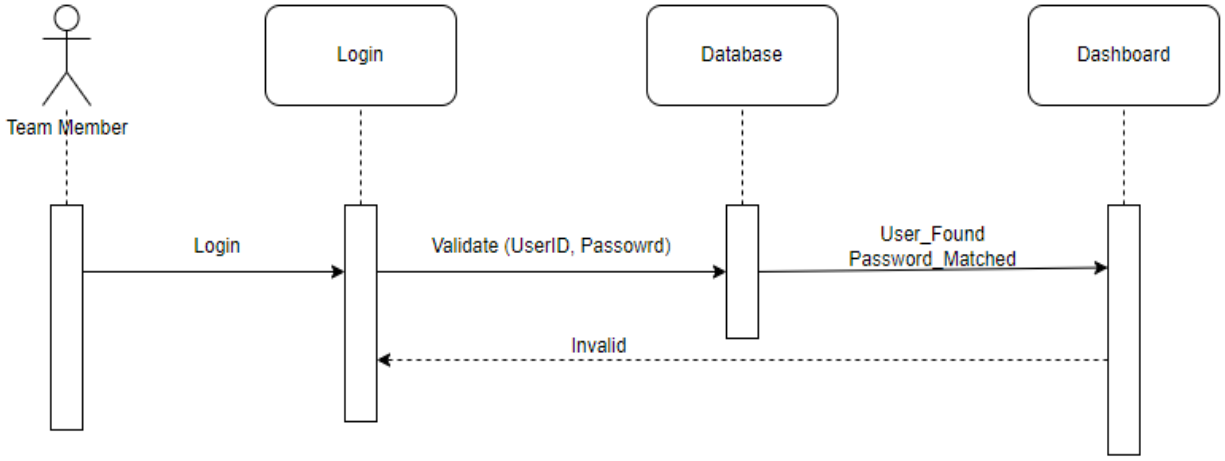
3. Control Object

Control objects are the parts of a software system that guarantee the system works properly and according to expectations. Control objects manage and regulate other objects' behaviour in the system. They are often included in the system architecture and are created to preserve the system's integrity, stability, and performance.

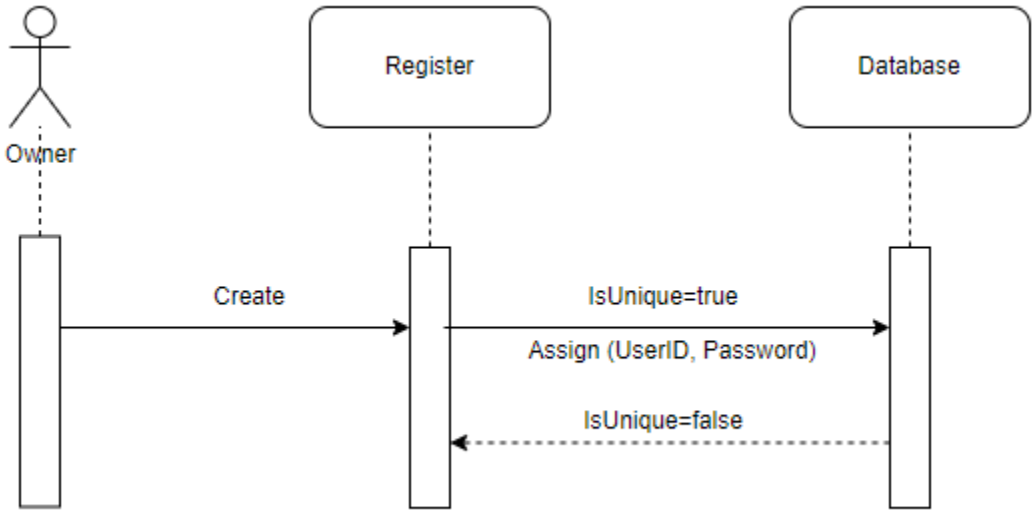
- **User Access Contro**l: The system should have role-based access control to ensure that each user has the appropriate level of access to the project management tool.
- **Task Management**: The tool should provide a centralized place for creating, assigning, and tracking tasks. The system should also allow for setting deadlines and reminders.
- **Project Management**: The system should enable users to create and manage projects with timelines, milestones, and deliverables.
- **Time Tracking**: The tool should have a time tracking feature that allows team members to log their hours worked on a specific task or project.
- **Resource Management**: The system should enable users to manage resources such as people, equipment, and materials required for a project.
- **Communication Management**: The system should enable users to communicate effectively within the team through email notifications, messaging, and comments.
- **Reporting and Analytics**: The tool should have reporting and analytics capabilities that allow users to track project progress, identify bottlenecks, and make data-driven decisions.

Sequence Diagram

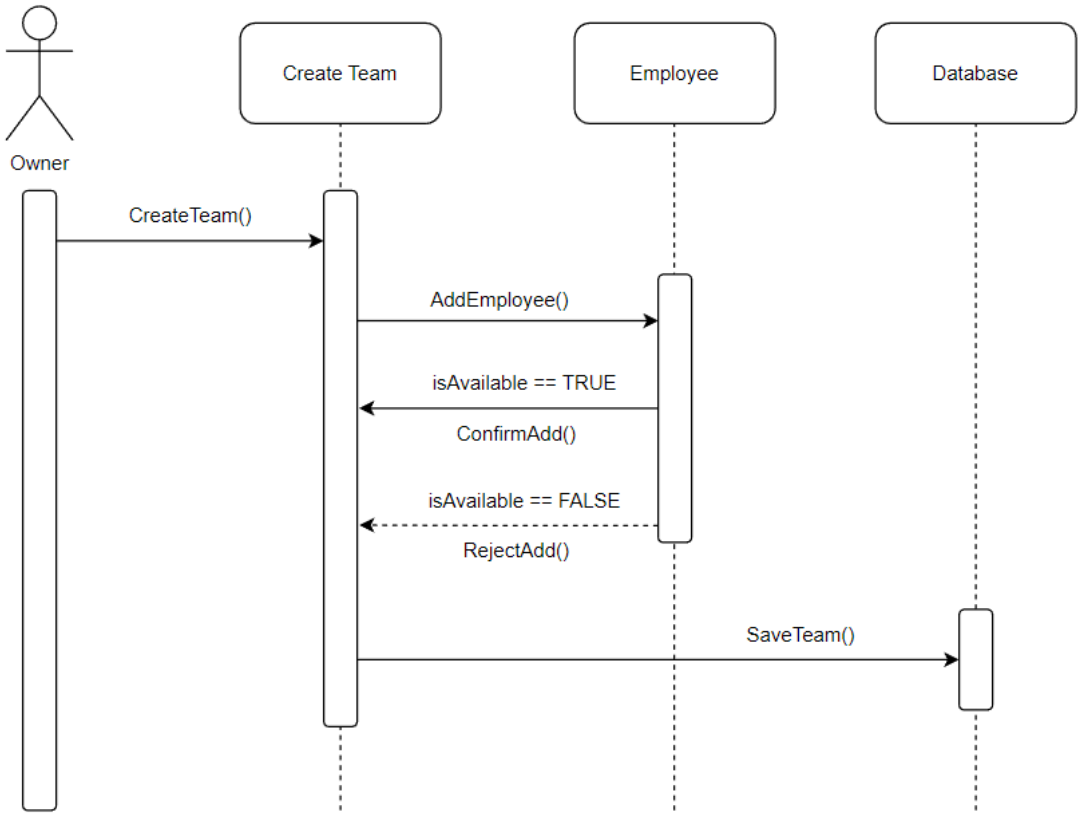
1. Login



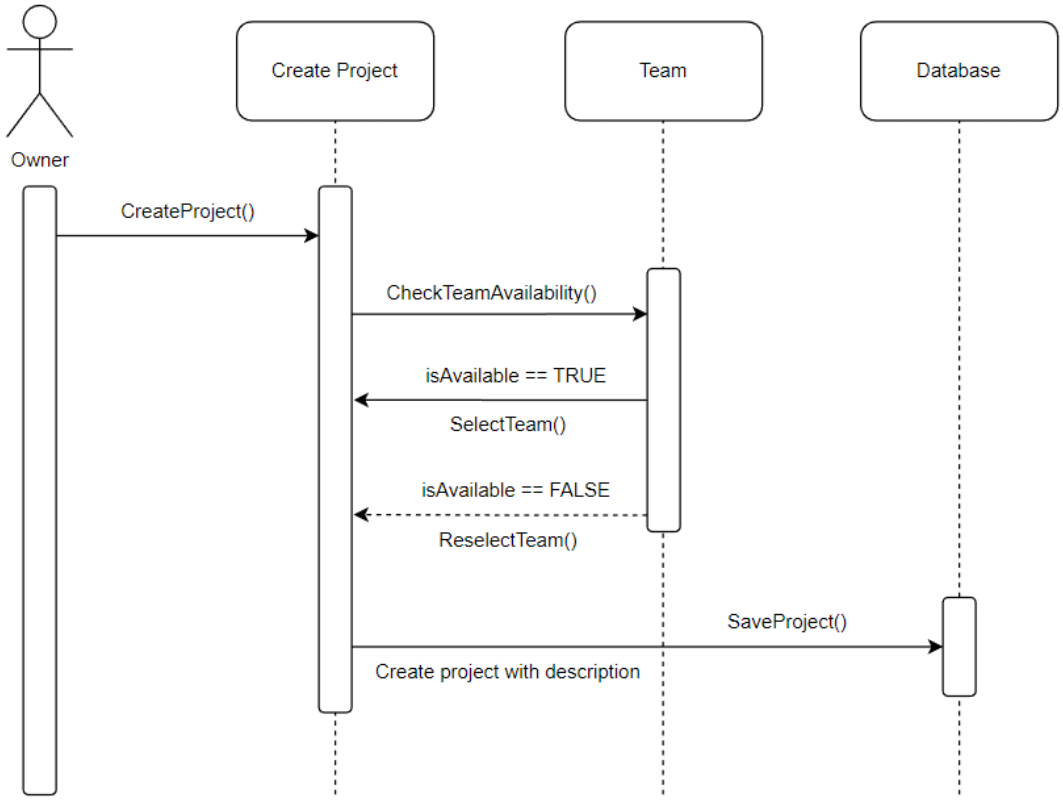
2. Create User



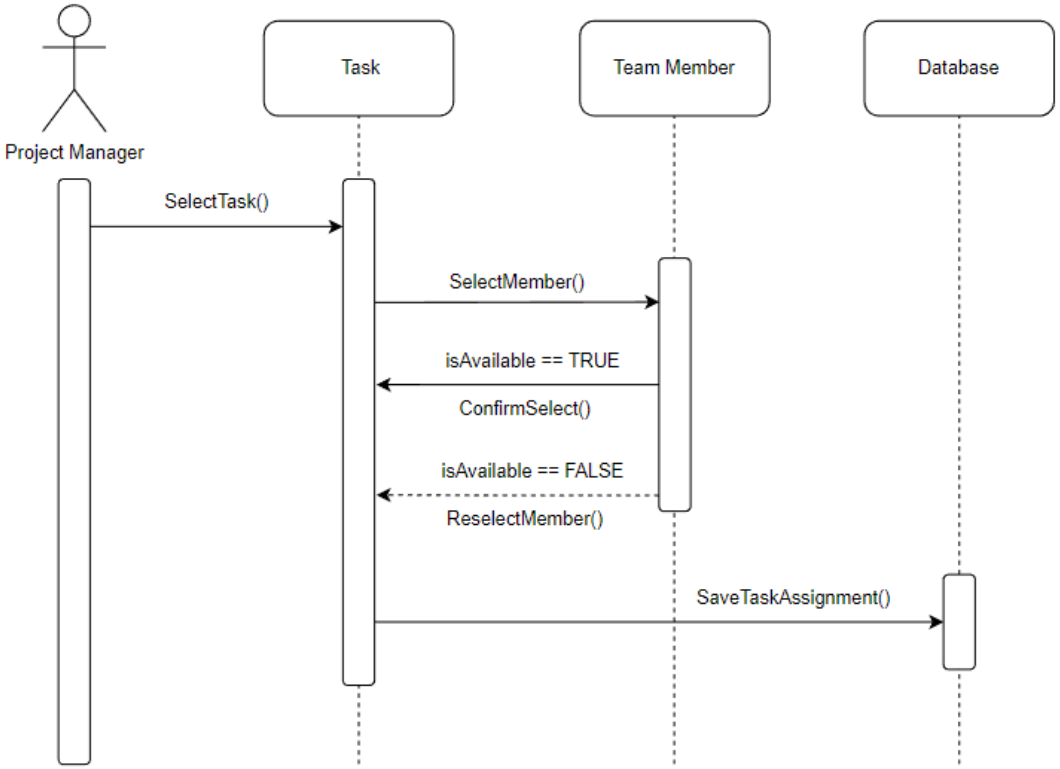
3. Create Team



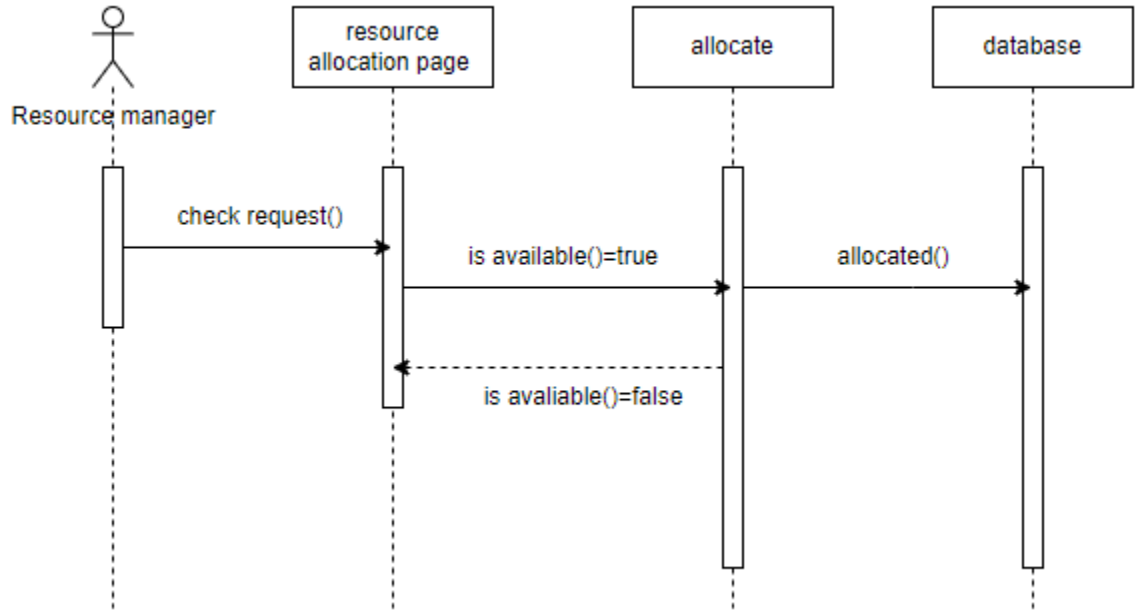
4. Create Project



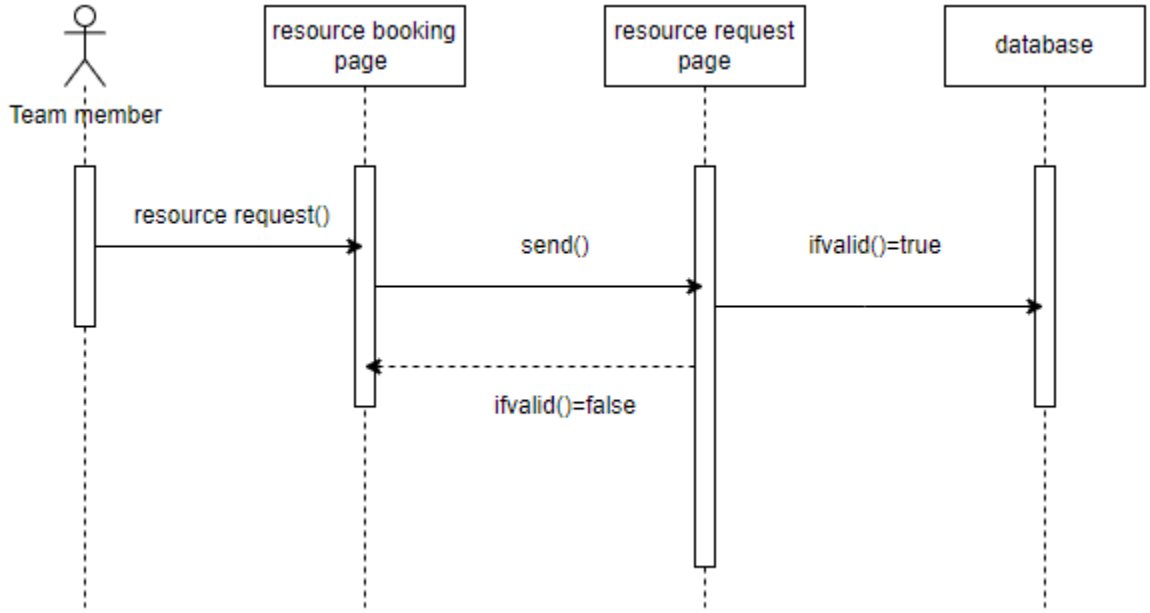
5. Assign Task



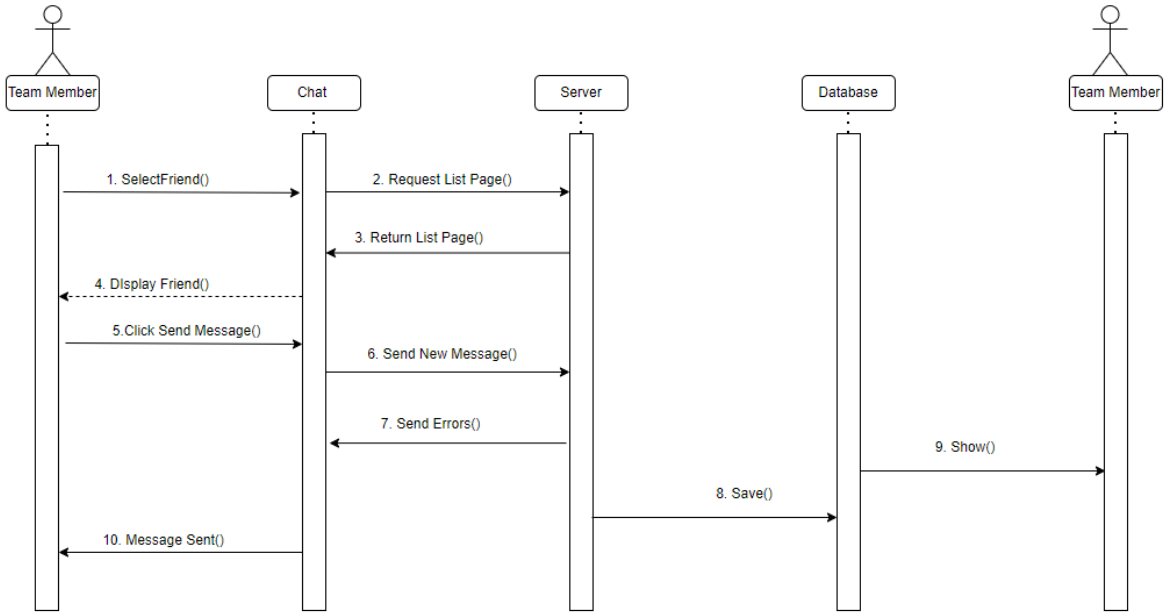
6. Resource Allocation



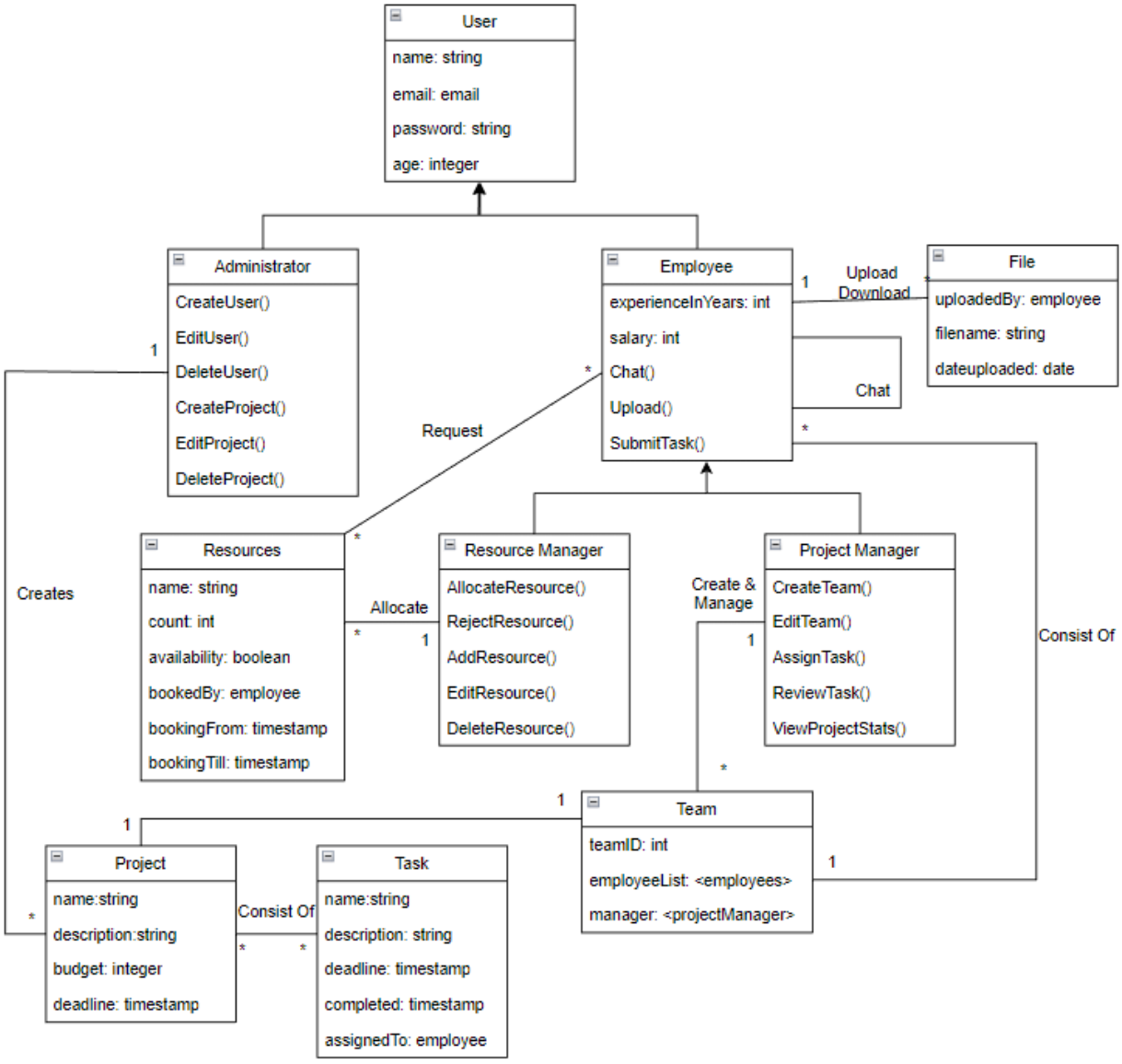
7. Resource Request



8. Chat


Team Member
Chat
Server
Database
Team Member
1. SelectFriend()
2. Request List Page()
3. Return List Page()
4. DIsplay Friend()
5.Click Send Message()
6. Send New Message()
7. Send Errors()
9. Show()
8. Save()
10. Message Sent()

Class Diagram



Identifying Design Goals

1. Functionality: Our system should be able to perform different functions like search recipes and its subcategories and general functionaries of Dashboard. Also, the sign-in/sign-up options work correctly as they are designed to do so.
2. Maintainability: The system will use the functional react components for maintenance. This will make the system highly reusable and ensure the components' independence.
3. Testing: To make the system more reliable and error-free, the system will be tested through static code analysis tools, which will help the admin/developer to identify minor bugs and make the code more readable. Also, the Testing Team will use various test cases to test in every corner of the system.
4. Ease Of Use: The system's overall user interface will be very handy for the user to use the system's functionalities. This will also reflect functionalities that involve editing and modifying the user's data.
5. Minimum number of errors: The system aims to make minimum errors in providing the results the user expects. Also, in a scenario where multiple requests are coming to the system, the system should be able to fetch the correct results for all the requests.
6. Reliability: The system will be designed in such a way that it will be able to handle multiple requests at a time and provide expected results.

High Level System Design

1. Architecture

This application will use a Model-View-Controller architecture. The model will act as the interface of data. It is responsible for maintaining data and is the logical data structure behind the entire application and is represented by a database.

The View is the user interface — what you see in your browser when a site renders. It is represented by HTML/CSS/Javascript and files.

A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

There are 3 main components or code partitions; Input Logic, Business Logic, and UI Logic. These can be divided into network layer tiers as follows.

Presentation tier: This layer is distributed to a computing device using a web browser or a web-based application and is constructed with HTML/CSS/JS. This component contains the UI logic in the Django architecture. View is actually the User Interface of the web-application and contains the parts like HTML, CSS and other frontend technologies.

Application tier: The application tier, also called the logic tier, is written in Python and contains the business logic that supports the application's core functions. The Django framework provides high security and reliability for writing business logic.

Data tier: A database and software for controlling read and write access to a database make up the data tier. We will be using MongoDB for the same, a NoSQL database. MongoDB will connect to Django via a python driver called PyMongo.

2. Identifying Sub-System