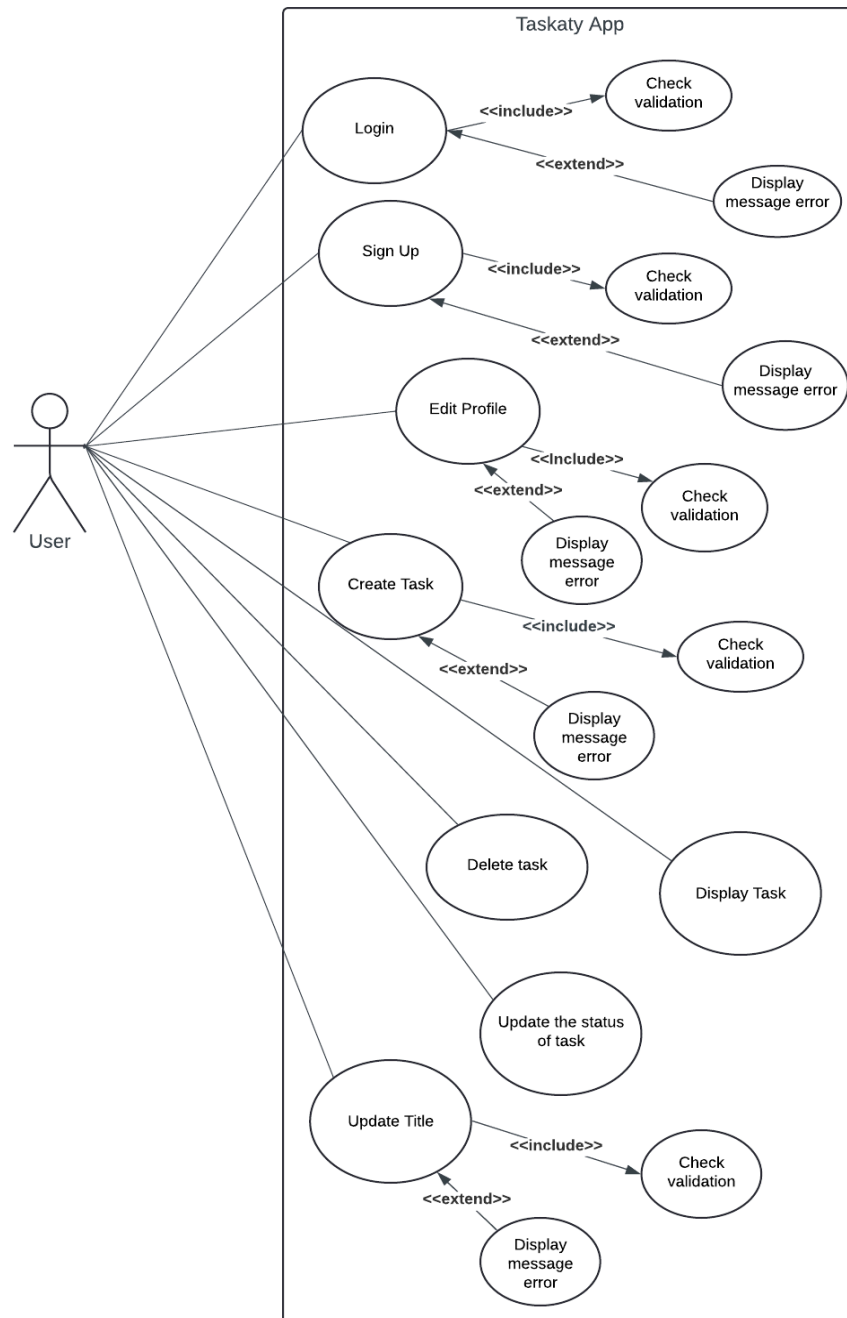
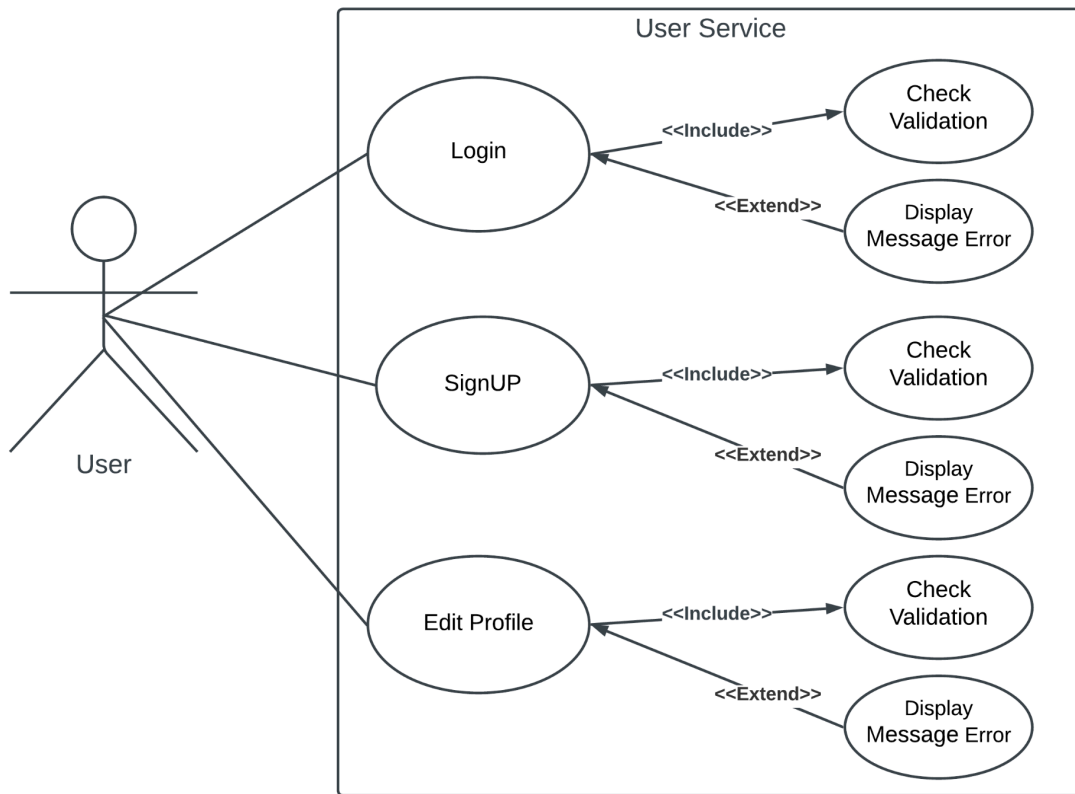


Software Requirements Specification (SRS)

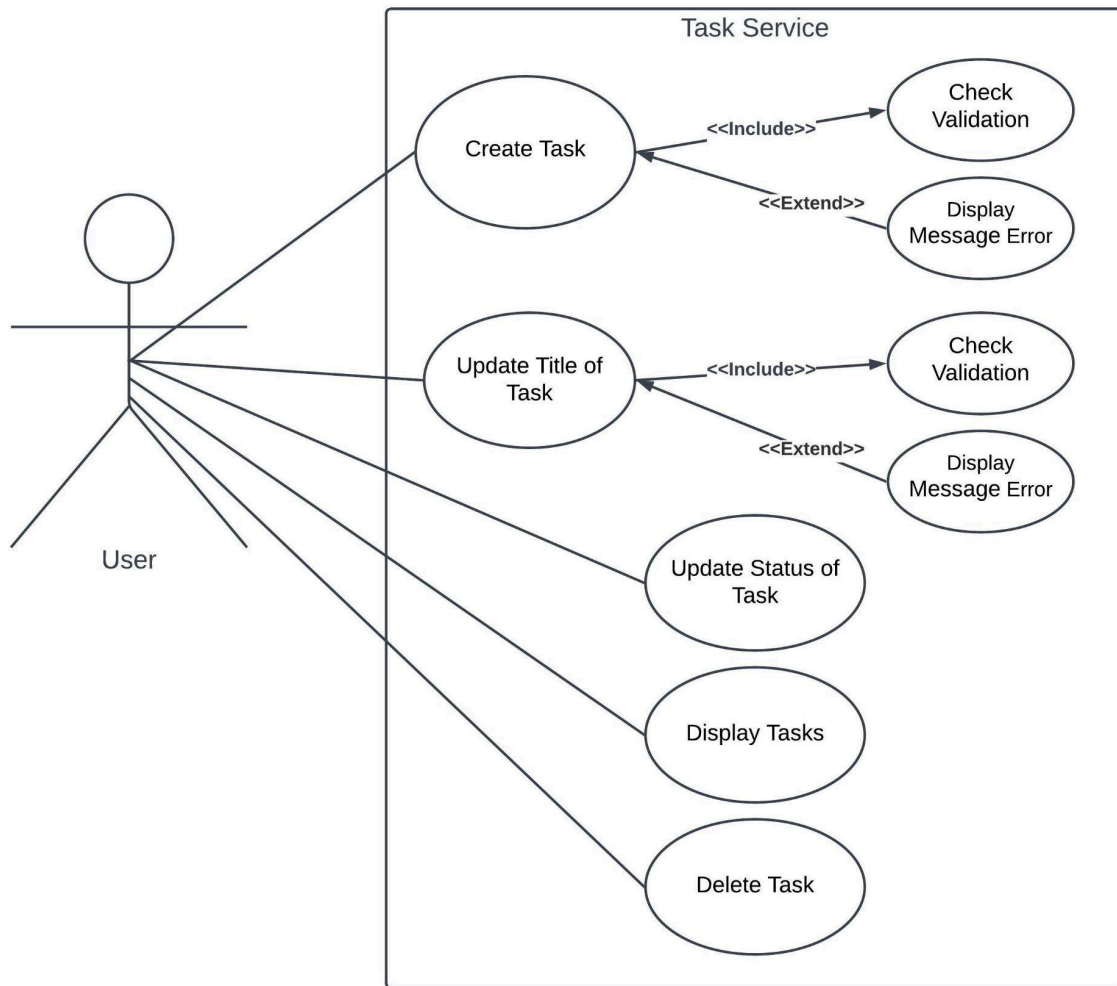
Use Case



User Service :



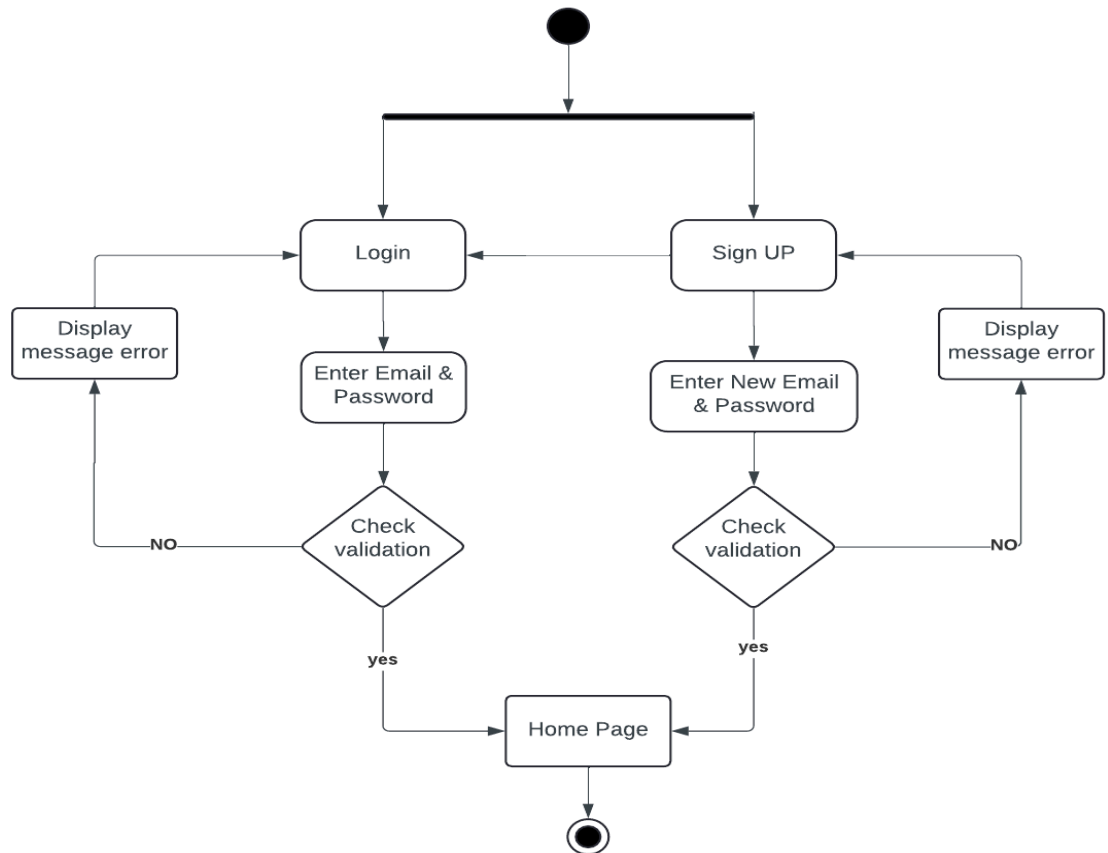
Task Service :



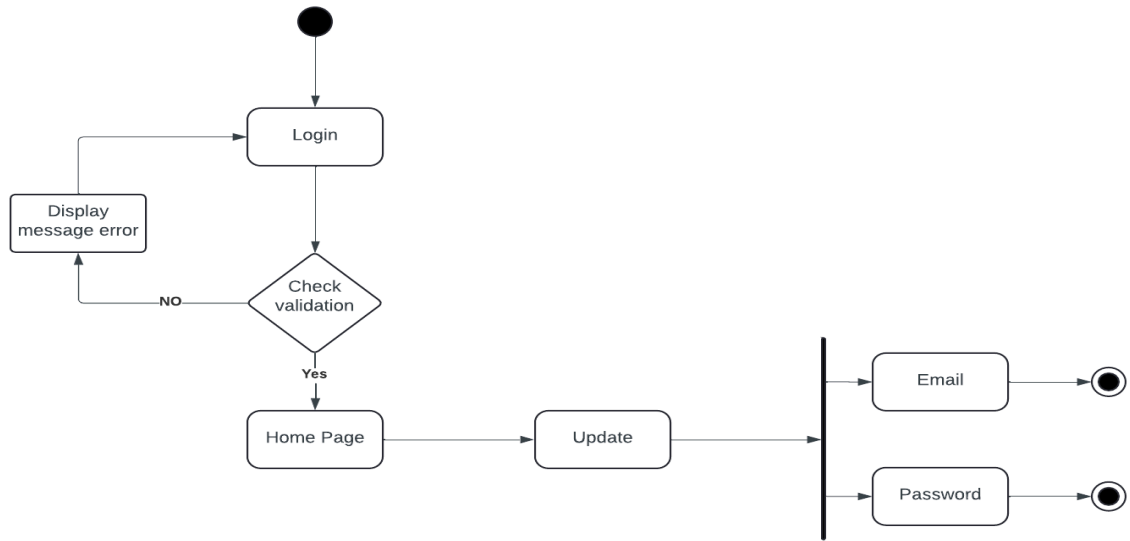
Activity diagram

User Service :

- Login & Signup

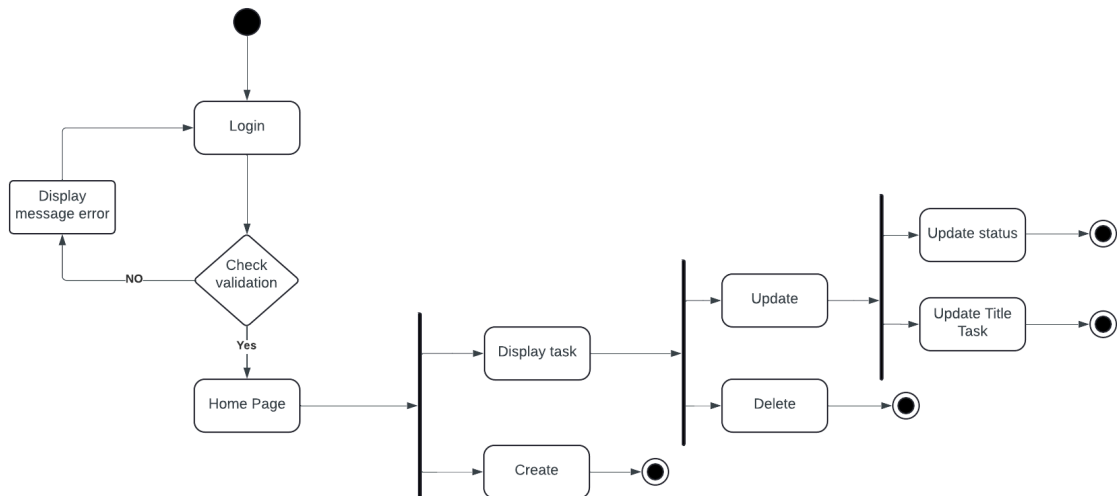


- **Edit Profile(Email,Password)**



Task Service :

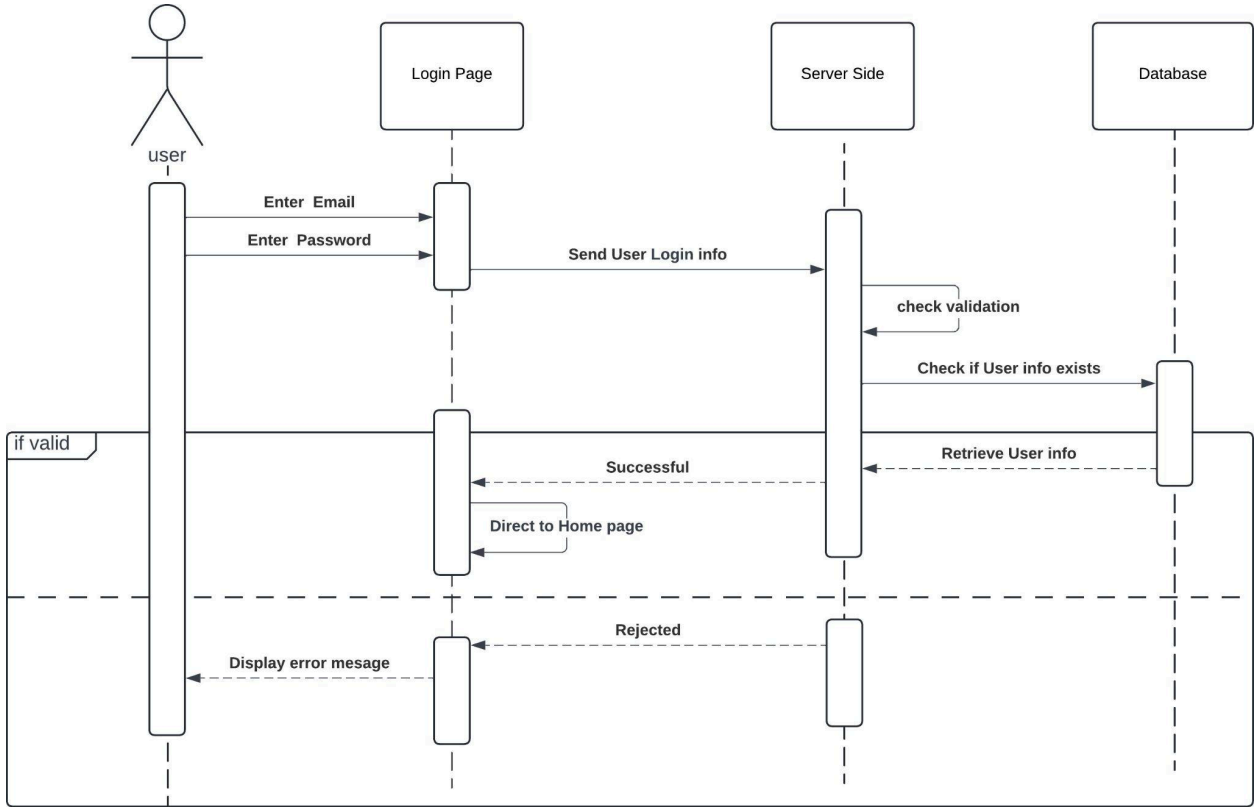
- **Task Management (Create - Display - Update - Delete)**



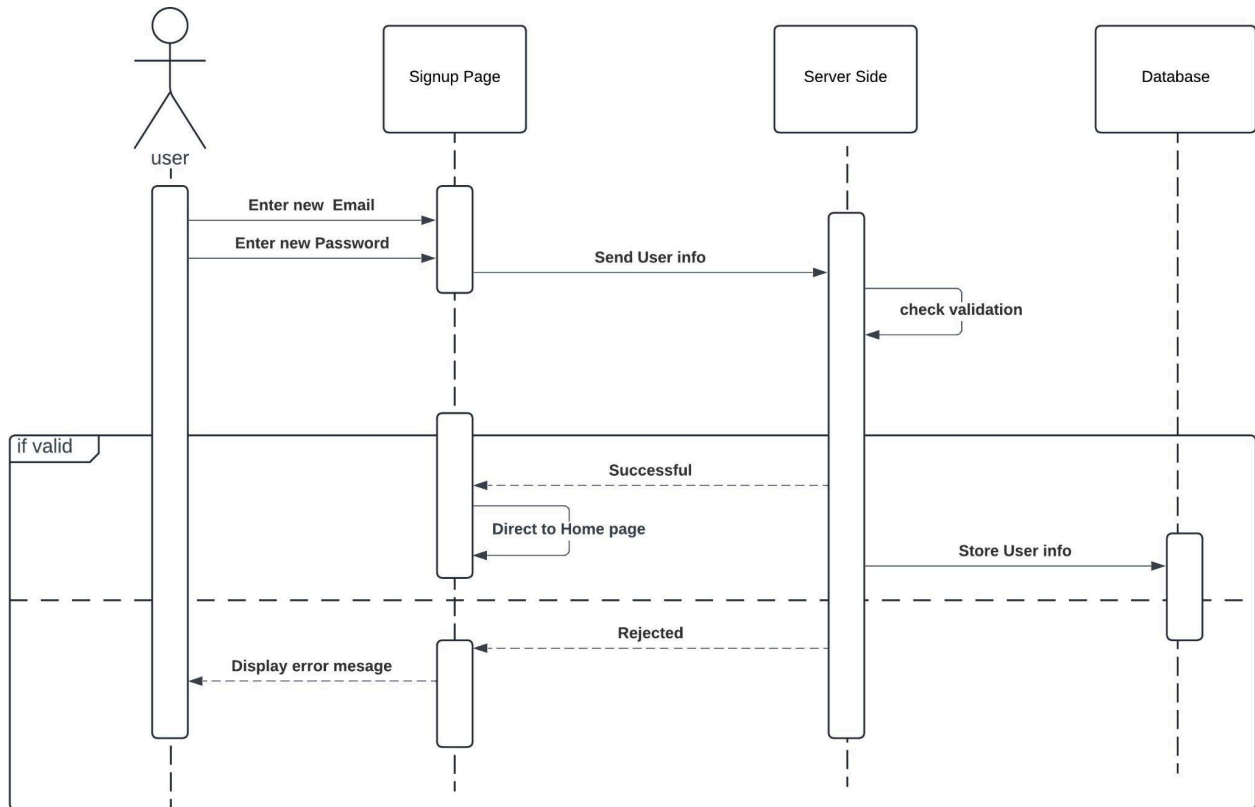
Sequence Diagram

User Service :

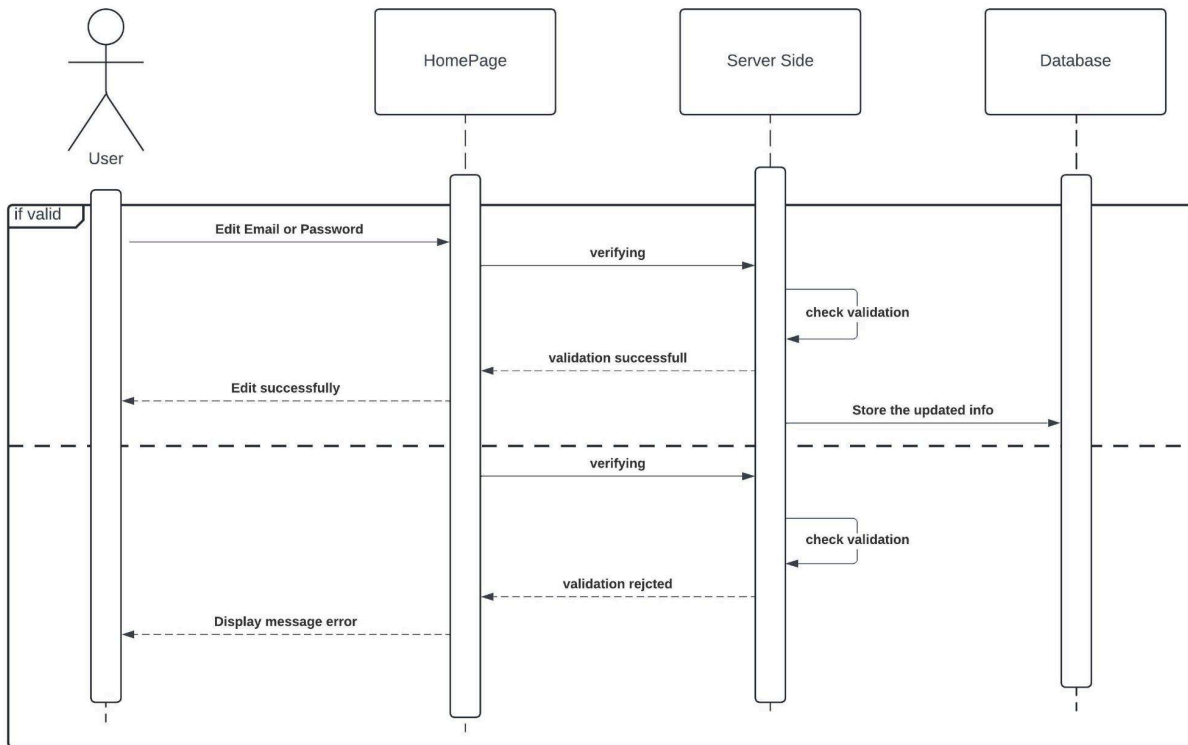
Login :



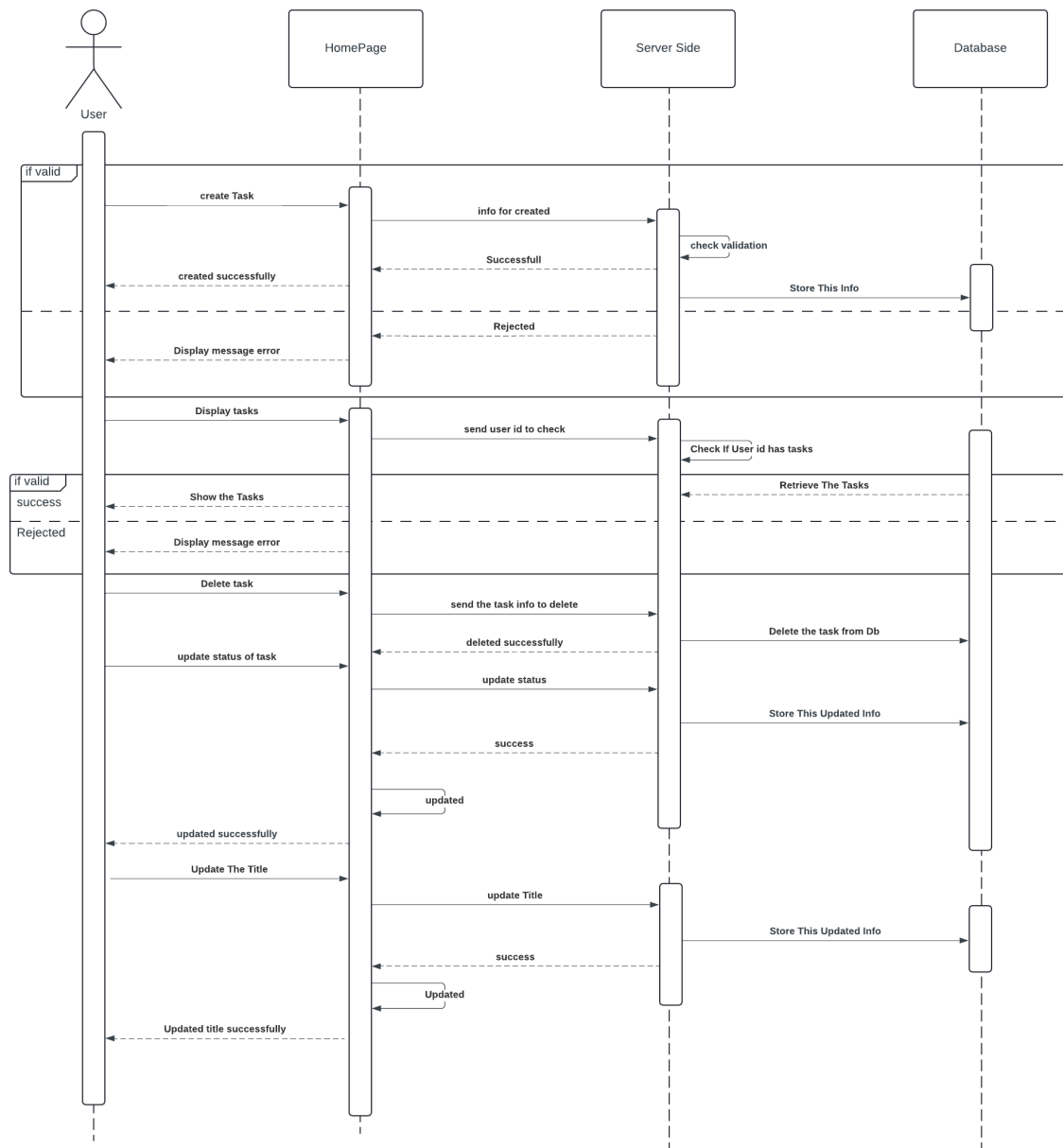
Sign Up :



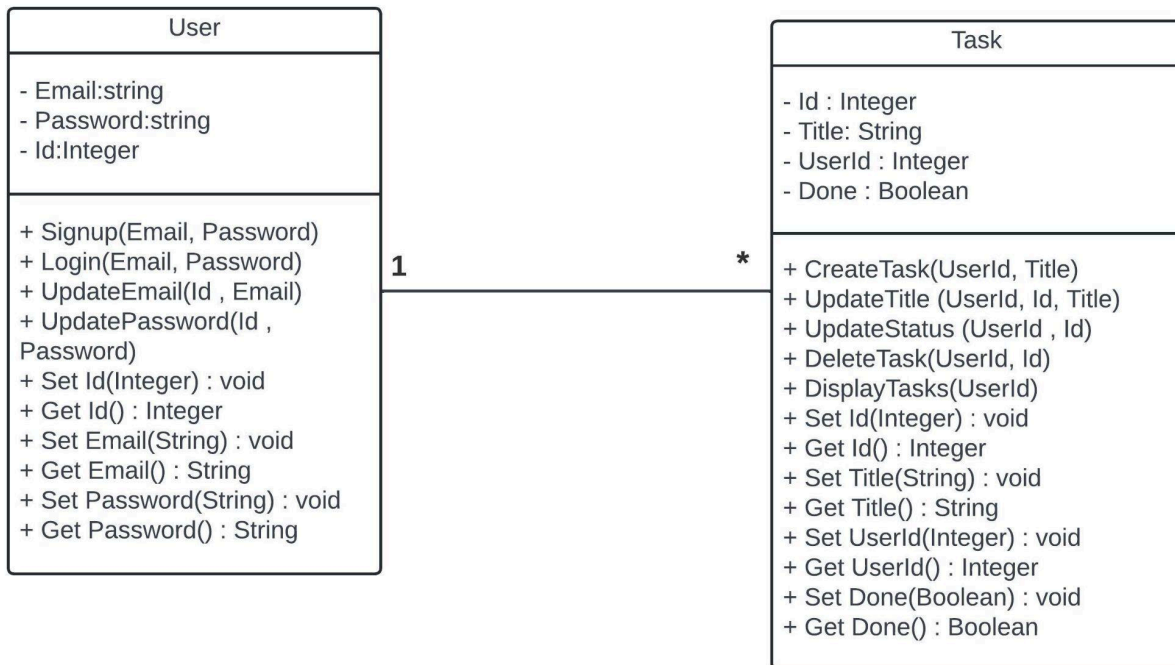
Edit Profile :



Task Service (Create - Display - Update - Delete) :



Class Diagram



OCL Diagram

Task Microservice:

Constraint 1: Tasks must have a title.

context Task

invariant: `self.title <> null and self.title.isEmpty()`

Constraint 2: Task titles must be between 1 and 255 characters long.

context Task

invariant: `self.title.size() >= 1 and self.title.size() <= 255`

Constraint 3: Tasks must have a user ID associated with them.

context Task

invariant: self.userId <> null

Constraint 4: Tasks must have a Status.

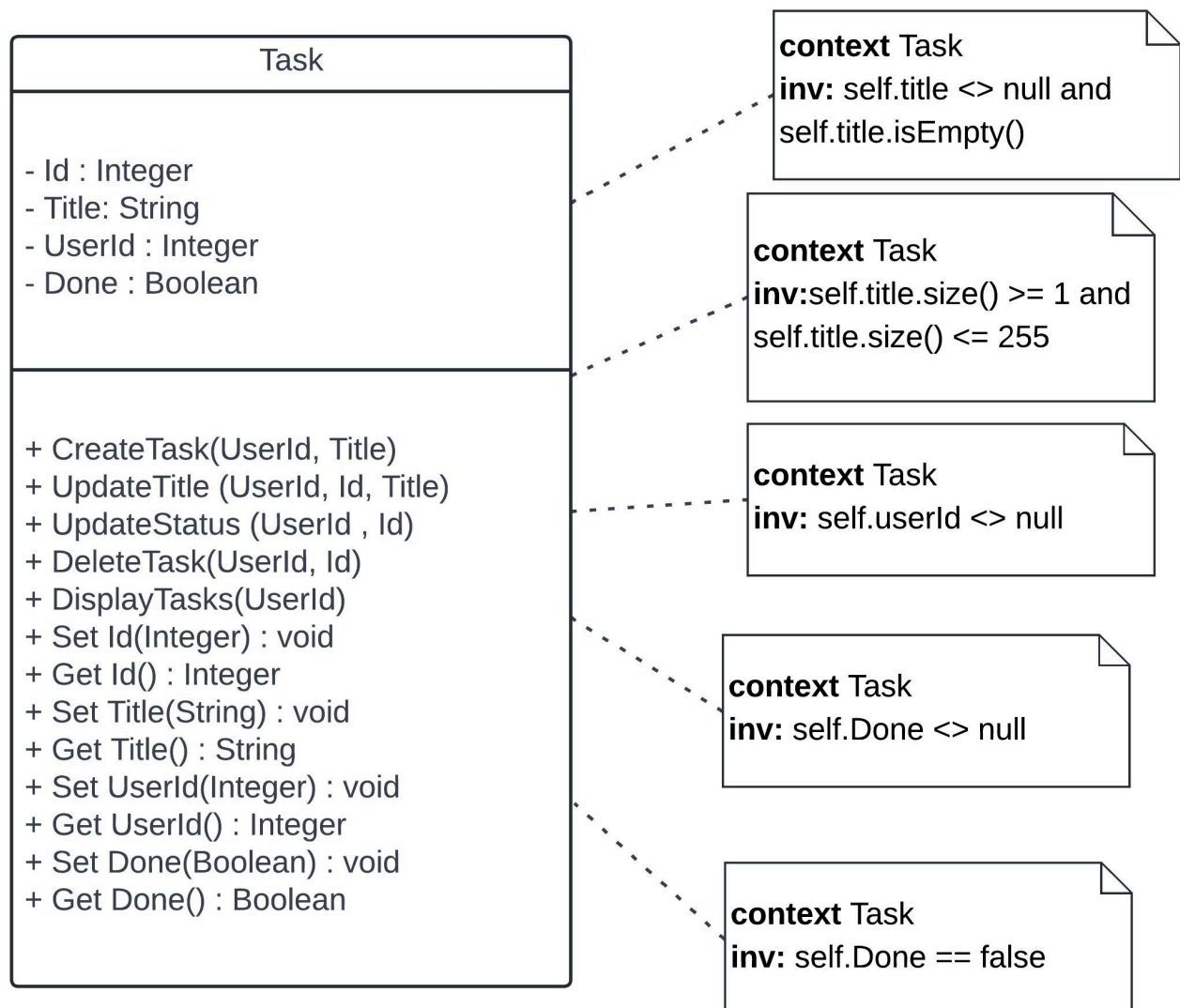
context Task

invariant: self.Done <> null

Constraint 5: Tasks when created must have a default status value.

context Task

invariant: self.Done == false



User Microservice:

Constraint 1: User email addresses must be unique within the system.

context User

invariant: `User.allInstances()->forall(u1, u2 | u1 <> u2 implies u1.email <> u2.email)`

Constraint 2: User email addresses must not be blank.

context User

invariant: `self.email <> null and self.email.isEmpty()`

Constraint 3: User email addresses must follow the standard email format.

context User

invariant: `self.email.matches('[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}')`

Constraint 4: User Password must not be blank.

context User

invariant: `self.password <> null and self.password.isEmpty()`

Constraint 5: User Password size must be between (6-20).

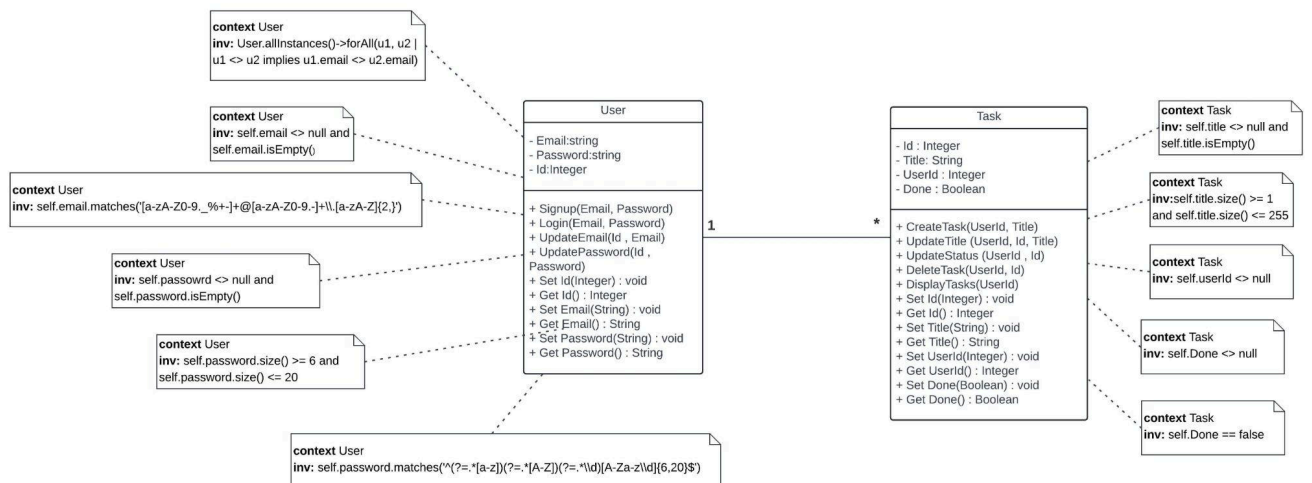
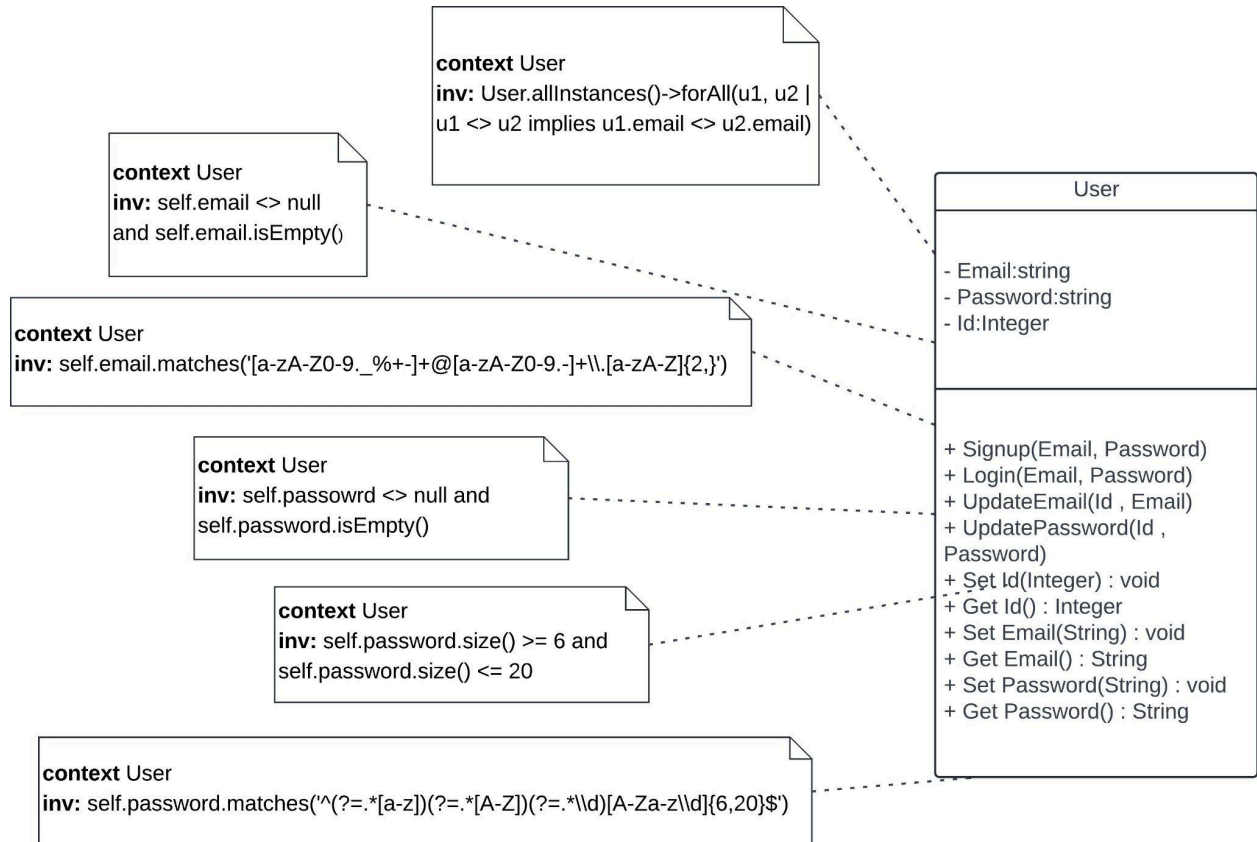
context User

invariant: `self.password.size() >= 6 and self.password.size() <= 20`

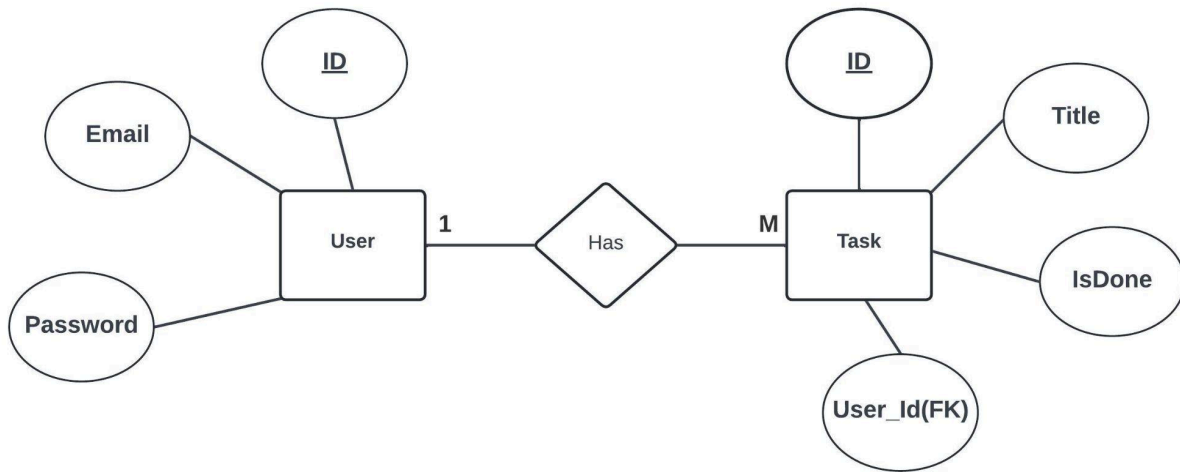
Constraint 6: User Passwords must meet some complexity requirements and regular expressions.

context User

invariant: `self.password.matches('^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[A-Za-z\d]{6,20}$')`



ERD



Entity Relational Schema

