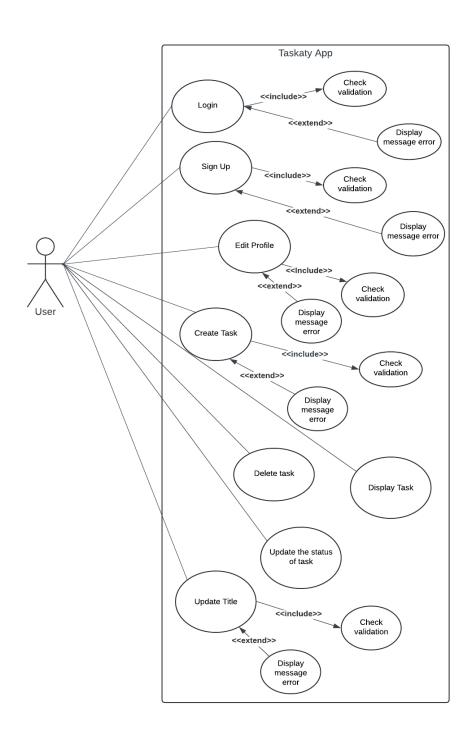
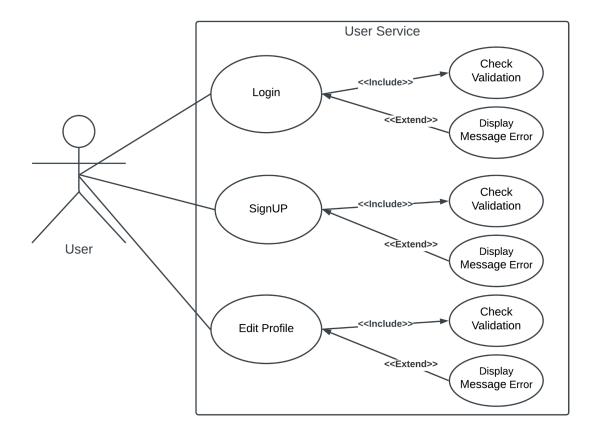
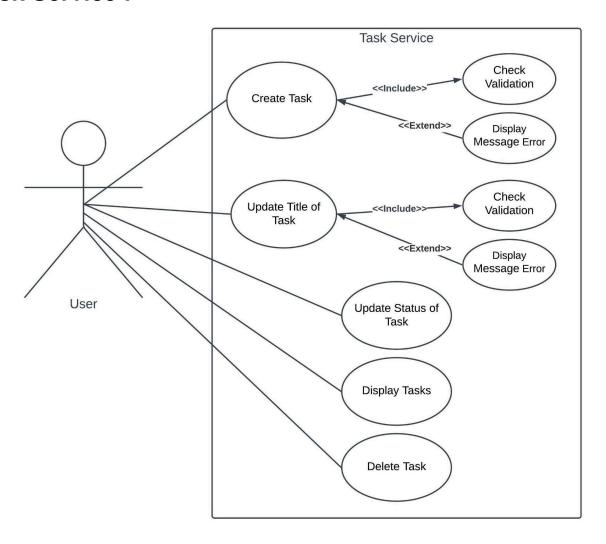
Software Requirements Specification (SRS) <u>Use Case</u>



User Service:



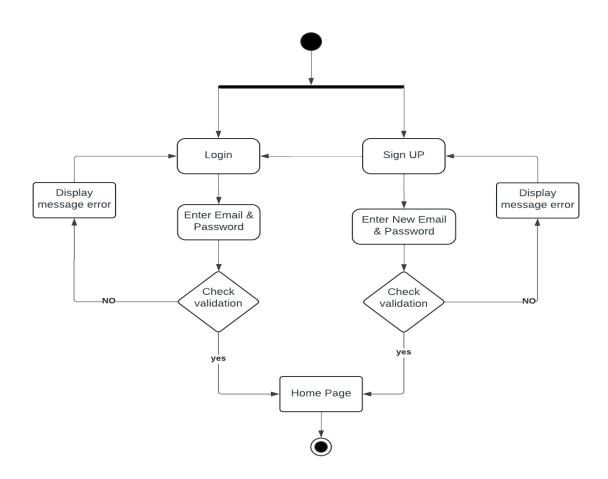
Task Service:



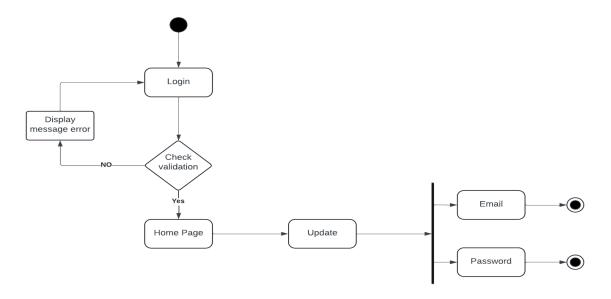
Activity diagram

User Service:

• Login & Signup

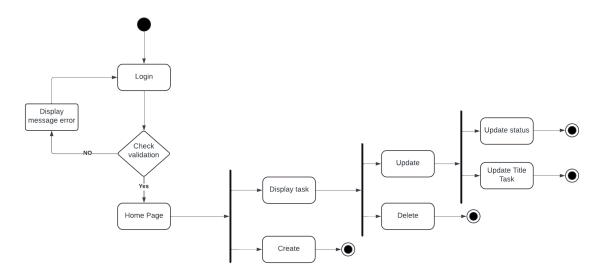


• Edit Profile(Email,Password)



Task Service:

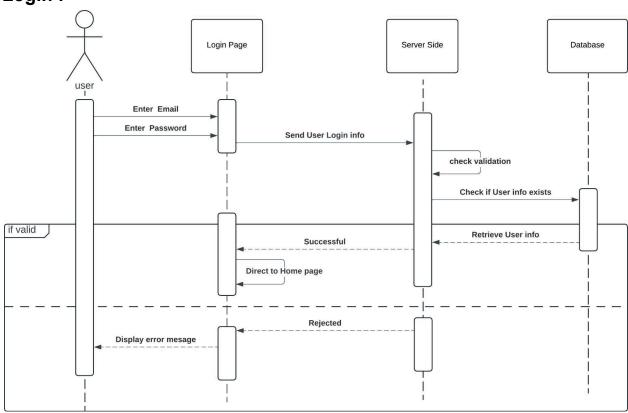
Task Management (Create - Display - Update - Delete)



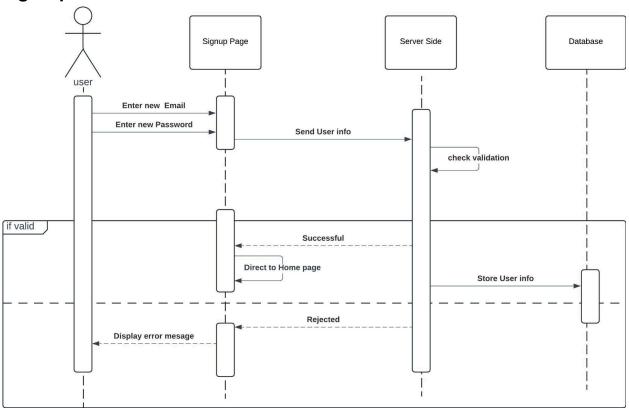
Sequence Diagram

User Service:

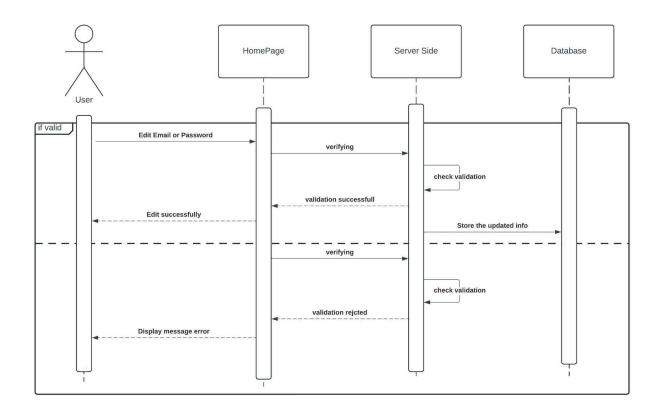
Login:



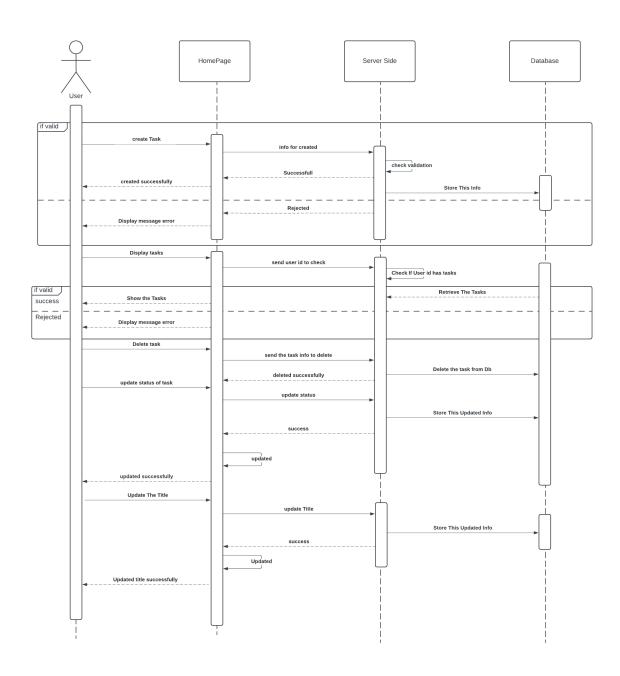
Sign Up:



Edit Profile:



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



Class Diagram

User		
		Task
- Email:string - Password:string - Id:Integer		- Id : Integer - Title: String - UserId : Integer - Done : Boolean
+ Signup(Email, Password) + Login(Email, Password) + UpdateEmail(Id, Email) + UpdatePassword(Id, Password) + Set Id(Integer): void + Get Id(): Integer + Set Email(String): void + Get Email(): String + Set Password(String): void + Get Password(): String	1 0*	+ CreateTask(UserId, Title) + UpdateTitle (UserId, Id, Title) + UpdateStatus (UserId, Id) + DeleteTask(UserId, Id) + DisplayTasks(UserId) + Set Id(Integer): void + Get Id(): Integer + Set Title(String): void + Get Title(): String + Set UserId(Integer): void + Get UserId(): Integer + Set Done(Boolean): void + Get Done(): Boolean

OCL Diagram

Task Microservice:

Constraint 1: Tasks must have a title.

context Task

invariant: self.title <> null

Constraint 2: Task titles must be between 1 and 100 characters long.

context Task

invariant: self.title.size() >= 1 and self.title.size() <= 100</pre>

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

context Task

invariant: self.userld <> null

Constraint 4: Tasks must have a Status.

context Task

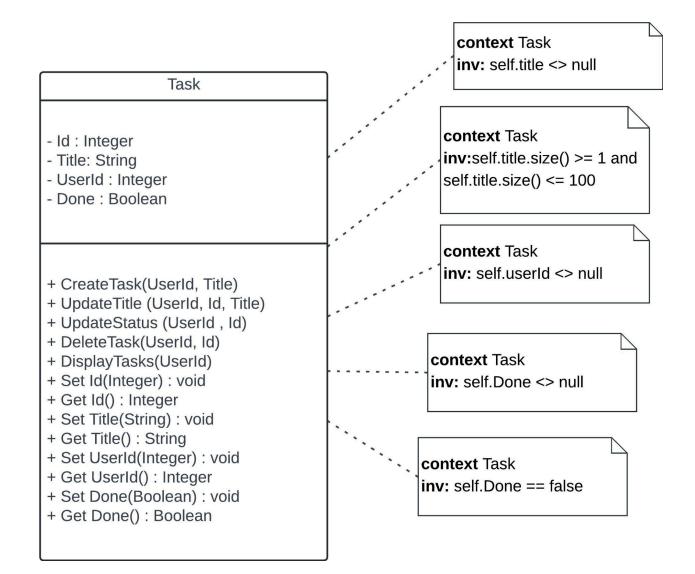
invariant: self.Done <> null

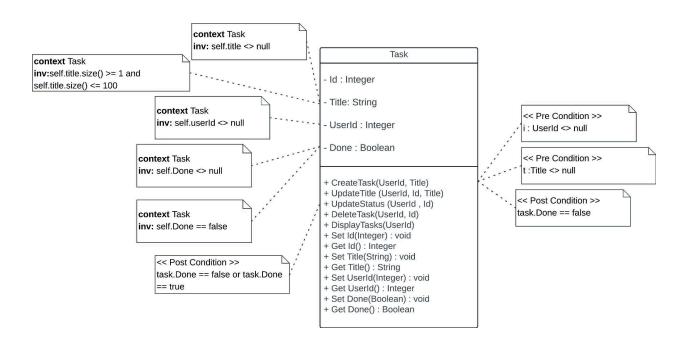
Constraint 5: Tasks when created must have a default status

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 must have an email address.

context User

invariant: self.email <> null

Constraint 3: User email addresses must follows 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 must have a Password.

context User

invariant: self.passowrd <> null

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

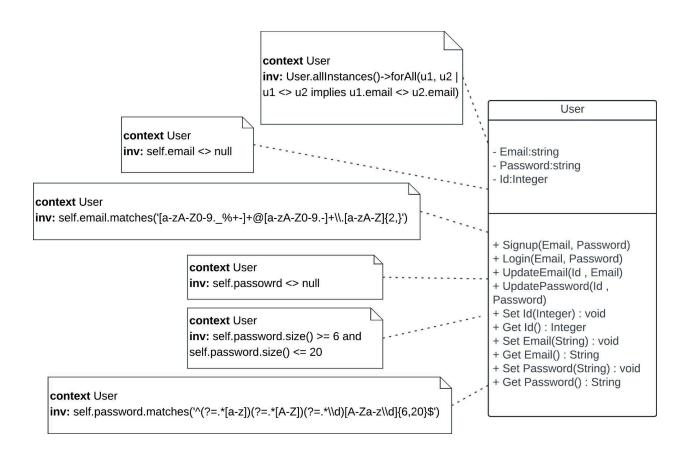
context User

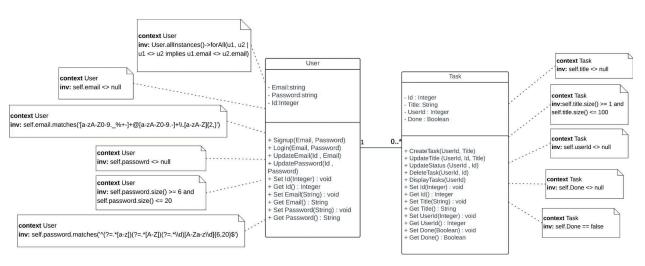
invariant: self.password.size() >= 6 and self.password.size() <= 20</pre>

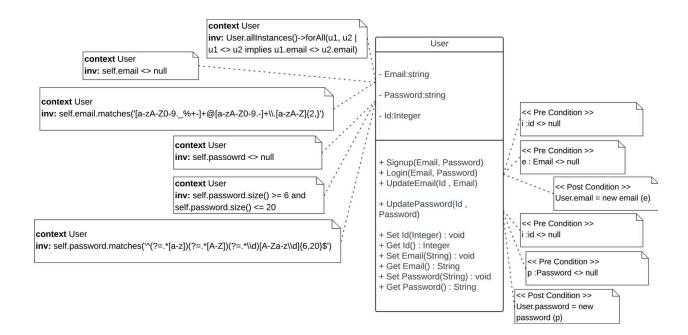
Constraint 6: User Passwords must meet some complexity requirements and regular expressions.

context User

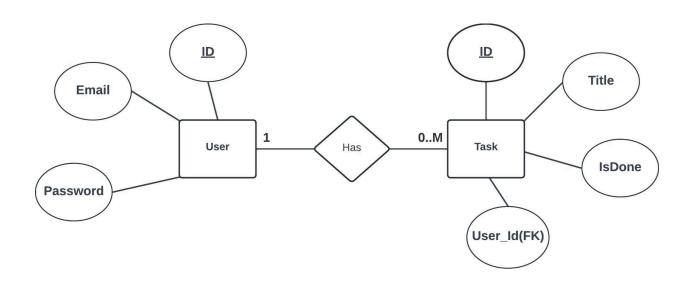
invariant:self.password.matches(' $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[A-Za-z\d]{6,20}$')$







ERD



Entity Relational Schema(DB Schema)

