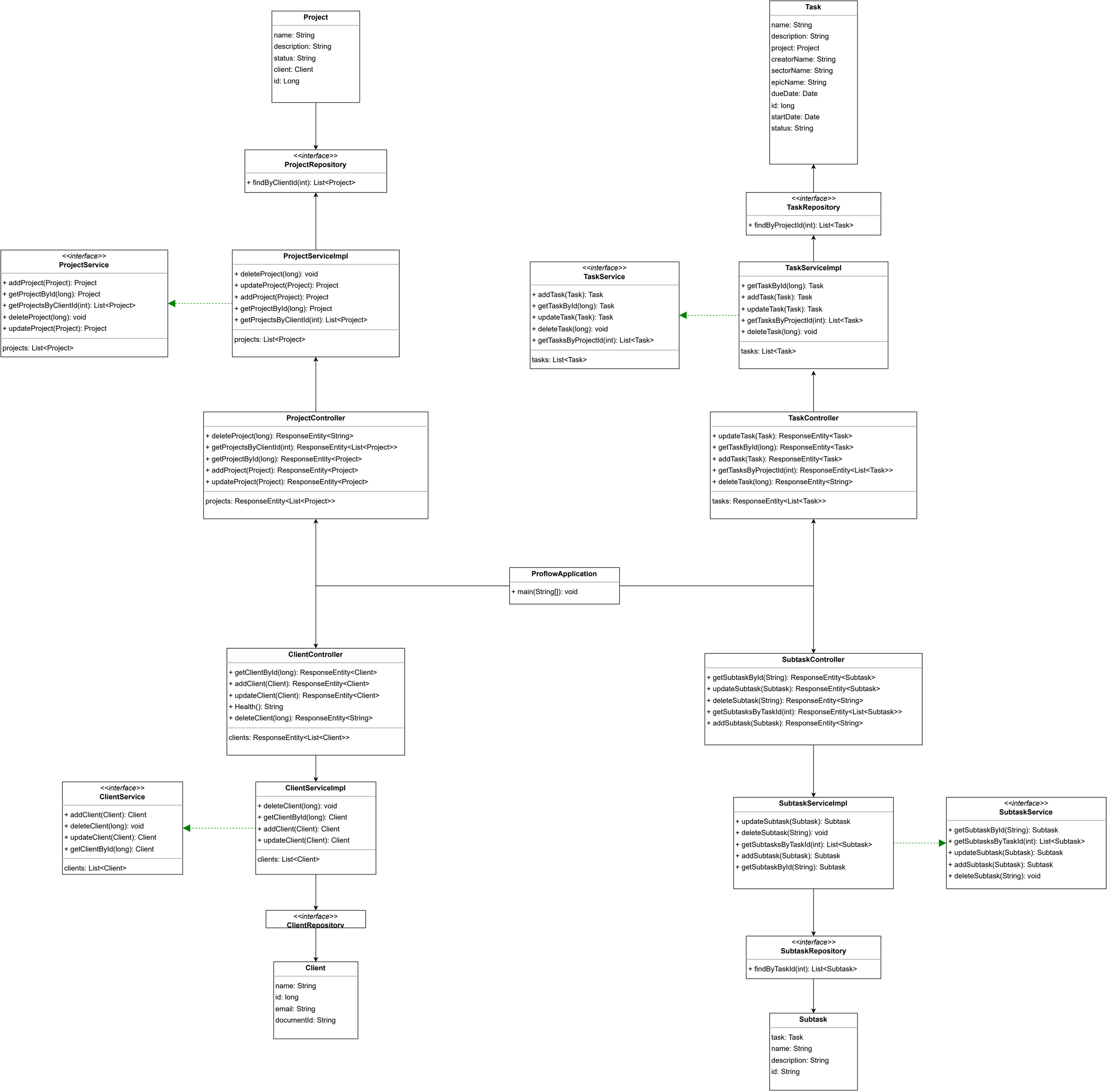
Project
name: String
description: String
status: String
client: Client
id: Long
<<interface>>
ProjectRepository
+ findByClientId(int): List<Project>
<<interface>>
ProjectService
+ addProject(Project): Project
+ getProjectById(long): Project
+ getProjectsByClientId(int): List<Project>
+ deleteProject(long): void
+ updateProject(Project): Project
projects: List<Project>
ProjectServiceImpl
+ deleteProject(long): void
+ updateProject(Project): Project
+ addProject(Project): Project
+ getProjectById(long): Project
+ getProjectsByClientId(int): List<Project>
projects: List<Project>
ProjectController
+ deleteProject(long): ResponseEntity<String>
+ getProjectsByClientId(int): ResponseEntity<List<Project>>
+ getProjectById(long): ResponseEntity<Project>
+ addProject(Project): ResponseEntity<Project>
+ updateProject(Project): ResponseEntity<Project>
projects: ResponseEntity<List<Project>>
Task
name: String
description: String
project: Project
creatorName: String
sectorName: String
epicName: String
dueDate: Date
id: long
startDate: Date
status: String
<<interface>>
TaskRepository
+ findByProjectId(int): List<Task>
<<interface>>
TaskService
+ addTask(Task): Task
+ getTaskById(long): Task
+ updateTask(Task): Task
+ deleteTask(long): void
+ getTasksByProjectId(int): List<Task>
tasks: List<Task>
TaskServiceImpl
+ getTaskById(long): Task
+ addTask(Task): Task
+ updateTask(Task): Task
+ getTasksByProjectId(int): List<Task>
+ deleteTask(long): void
tasks: List<Task>
TaskController
+ updateTask(Task): ResponseEntity<Task>
+ getTaskById(long): ResponseEntity<Task>
+ addTask(Task): ResponseEntity<Task>
+ getTasksByProjectId(int): ResponseEntity<List<Task>>
+ deleteTask(long): ResponseEntity<String>
tasks: ResponseEntity<List<Task>>
ProflowApplication
+ main(String[]): void
ClientController
+ getClientById(long): ResponseEntity<Client>
+ addClient(Client): ResponseEntity<Client>
+ updateClient(Client): ResponseEntity<Client>
+ Health(): String
+ deleteClient(long): ResponseEntity<String>
clients: ResponseEntity<List<Client>>
<<interface>>
ClientService
+ addClient(Client): Client
+ deleteClient(long): void
+ updateClient(Client): Client
+ getClientById(long): Client
clients: List<Client>
ClientServiceImpl
+ deleteClient(long): void
+ getClientById(long): Client
+ addClient(Client): Client
+ updateClient(Client): Client
clients: List<Client>
<<interface>>
ClientRepository
Client
name: String
id: long
email: String
documentId: String
SubtaskController
+ getSubtaskById(String): ResponseEntity<Subtask>
+ updateSubtask(Subtask): ResponseEntity<Subtask>
+ deleteSubtask(String): ResponseEntity<String>
+ getSubtasksByTaskId(int): ResponseEntity<List<Subtask>>
+ addSubtask(Subtask): ResponseEntity<String>
SubtaskServiceImpl
+ updateSubtask(Subtask): Subtask
+ deleteSubtask(String): void
+ getSubtasksByTaskId(int): List<Subtask>
+ addSubtask(Subtask): Subtask
+ getSubtaskById(String): Subtask
<<interface>>
SubtaskService
+ getSubtaskById(String): Subtask
+ getSubtasksByTaskId(int): List<Subtask>
+ updateSubtask(Subtask): Subtask
+ addSubtask(Subtask): Subtask
+ deleteSubtask(String): void
<<interface>>
SubtaskRepository
+ findByTaskId(int): List<Subtask>
Subtask
task: Task
name: String
description: String
id: String