Jalapenos Issue Tracker

The following diagram comprises the system design for the Issue Tracker system. The Entity models contain all the relevant data fields. The EntityService template class will handle saving and reading files related to each entity. EntityControllers will utilize EntityServices via dependency injection. T: Entity <<interface>> <<interface>> Entity EntityService<T> + Id: string # file: string -----<

</bind>>._____ + virtual toJson(): string + virtual get(): set<T> # generateId(): string + virtual get(int): T + virtual create(T): T implements + virtual update(T): T + virtual delete(T): bool Issue User Vote Comment implements **IssueService** + Title: string + Name: string + CreatedAt: tm + Body: string + Status: string + Role: string + CreatedBy: User + CreatedBy: User + AssignedTo: User + CreatedAt: tm CommentService + Reporter: User + UpdatedBy: User + Comments: set<Comment> + UpdatedAt: tm + CreatedAt: tm tm refers to L the ctime **VoteService** + UpdatedBy: User struct tm type + UpdatedAt: tm + Votes: set<Vote> **UserService** UserController <<interface>> EntityController + UserController(UserService) + endpoint: const string + resource: restbed::Resource VoteController **Uses** + VoteController(VoteService, UserService) + virtual getOne(Session*): void implements-+ virtual getAll(Session*): void CommentController + virtual create(Session*): void ------Uses-----+ CommentController(CommentService, UserService) + virtual update(Session*): void + virtual delete(Session*): void + error_handler(int, exception, Session*): void **IssueController** -Uses-+ IssueController(IssueService, CommentService, VoteService, UserService)