Spring MVC 3-level architecture Note: Every dependency without label here is: Manage Data(Service -> Data Access) Note: Each class in this architecture is a sinleton and the property is set automatically by Spring framework Data Service Layer Access Controller Layer AccountController AccountService Uses + getAllUsers(): List<User> + getAllUsers(): List<User> register(request:RegisterRequest): Result register(request:RegisterRequest): Result + registerAdmin(adminId:String, email:String, password:String): Result + login(email:String, password:String): Result registerAdmin(id:String, email:String, password:String): Result login(email:String, password:String): Result UserRepository + findByEmailaddress(emailaddress:String): User + findByStudentId(studentId:String):User **ArtifactService** + findUserByFirstname(firstname:String): List<User> ArtifactController + findUserByLastname(lastname:String): List<User> + showCategoryArtifactAttachments(email:String): + findUserByUsername(username:String): List<User> + showCategoryArtifactAttachments(): Map<String, Map<String, List<String>>> + upload(email:String, category:String, title:String, description:String, Uses Map<String, Map<String, List<String>>> + findUserByLink(link:String): User + upload(email:String, category:String, title:String, description:String, attachment:List<String>, privacy:String): Result attachment:List<String>, privacy:String): Result +deleteArtifact(request: ViewArtifact): Result +deleteArtifact(request: ViewArtifact): Result +viewAttachment(request:ViewAttachmentRequest): ViewAttachmentResult +viewAttachment(request:ViewAttachmentRequest): ViewAttachmentResult +viewAttachmentByLink(link:String,category:String,artifact:String,attachment:String +switchPrivacy(email:String, category:String, artifact:String, privacy:String): Result +switchPrivacy(email:String, category:String, artifact:String, privacy:String): Result CacheService CacheController Uses upload(email:String, file:MultipartFile): Result CacheRepository - uploadCache(email:String, file:MultipartFile): Result + clear(email:String):Result - clearCache(email:String):Result getFileFromCache(email: String, files: List<String>):List<Attachment> + findByEmailaddress(emailaddress:String): UserCache + deleteByEmailaddress(emailaddress:String): Void + findAllByEmailaddress(emailaddress:String): List<UserCache> CategoryController CategoryService - addCategory(request:CategoryRequest): Result + addCategory(request:CategoryRequest): Result Uses + deleteCategory(request:CategoryRequest): Result - deleteCategory(request:CategoryRequest): Result + showCategories(email: String): AllCategoryResult + getAllCategory(email: String): AllCategoryRequest GuestService GuestController 1 + findUser(link: String): User Uses + showContent(link:String): Map<String, Map<String, List<String>>> + findPublicContect(user: User): Map<String, Map<String, List<String>>> + findPubByLink(link:String): Map<String, Map<String, List<String>>> + findProfileByLink(link:String): Profile + showProfile(link: String): Profile AdminRepository + findByEmailaddress(emailaddress:String): Administrator **ProfileController ProfileService** + findAllByAdminId(adminId:String): Administrator + updateAvatar(email:String, image:MultipartFile): Result Uses updateAvatar(email:String, image:MultipartFile): Result + deleteAvatar(email:String): Result deleteAvatar(email:String): Result + getProfile(email:String): Profile + getUserProfile(email:String): Profile + updateProfile(email:String, firstname:String, lastname:String, username:String,description:String): Result + updateProfile(email:String, firstname:String, lastname:String, username;String, generateLink(email:String): String description: String): Result + generateLink(email:String): String SearchService SearchController Uses + showUser(info: String): List<User> + showUser(info: String): Map<String, List<String>> - addToUsers(users: List<User>, user: User): void

