**Traceability diagram:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Packages** | **Class** | **Method** |
| Create Model | ui | Main | createController() |
| model | Controller | Controller() |
| Req#1: register a community | ui | Main | registerCommunity() |
| model | Controller | searchCommunity(name:String) |
| model | Community | getName() |
| model | Controller | availableCommunity() |
| model | Controller | displayCommunityTypes() |
| model | Community | getCommunityTypes() |
| model | CommunityType | getCommunityTypes() |
| model | CommunityType | CommunityType(description: String) |
| model | CommunityType | getDescription() |
| model | Controller | displayChallengeTypes() |
| model | Community | getChallengesString() |
| model | Challenge | getChallengesString() |
| model | Challenge | Challenge(description : String) |
| model | Challenge | getDescription() |
| model | Controller | registerCommunity(name : String, representantName : String, representantPhone : String, population : int, hospitals : String, schools : Strings, water: String, food : String, intType : int) |
| model | Community | Community(name : String, representantName : String, representantPhone : String, population : int, chosenChallenges : int[], intType : int) |
| model | Challenge | intToChallenge(intChallenge : int) |
| model | CommunityType | intToCommunityType(intType: int) |
| Req#2: Register a place | ui | Main | registerPlace() |
| model | Controller | displayCommunities() |
| model | Controller | oneMinCommunity() |
| model | Community | getName() |
| model | Controller | intToCommunity() |
| model | Controller | displayPlaceTypes() |
| model | Place | getPlaceTypes() |
| model | PlaceType | getPlaceTypes() |
| model | PlaceType | PlaceType(description: String) |
| model | PlaceType | getDescription() |
| model | Controller | displayDepartments() |
| model | Place | getDepartments() |
| model | Department | getDepartments() |
| model | Department | Department(description: String) |
| model | Department | getDescription() |
| model | Controller | searchPlace(name:String) |
| model | Place | getName() |
| model | Controller | availablePlace() |
| model | Controller | registerPlace(intCommunity: int, placeName : String, area : double, inaugurationDate : Calendar, photo : String, resources : double, intDepartment : int, intType : int) |
| model | Place | Place(name : String, area : double, inaugurationDate : Calendar, photo : URL, resources : double, department : Department, type : PlaceType, caregivingCommunity: Community) |
| model | PlaceType | intToPlaceType(intType : int) |
| model | Department | intToDepartment(intDepartment : int) |
| Req#3: Add a product to a community | ui | Main | addProduct() |
| model | Controller | oneMinCommunity() |
| model | Controller | displayCommunities() |
| model | Community | getName() |
| model | Controller | displayProductTypes() |
| model | Community | getProductTypes() |
| model | Product | getProductTypes() |
| model | ProductType | getProductTypes() |
| model | ProductType | PlaceType(description: String) |
| model | ProductType | getDescription() |
| model | Controller | addProduct(intCommunity : int, productName : String, naturePercent : double, handcraft : String, intType : int) |
| model | Controller | intToCommunity() |
| model | Product | Product(name : String, naturePercent : double, handcraft : String, intType: int) |
| Model | ProductType | intToProductType(intType : int) |
| model | Community | addProduct(newProduct : Product) |
| model | Community | searchProduct(productName : String) |
| model | Product | getName() |
| model | Community | availableProduct() |
| Req#4: Delete a product from a community | ui | Main | deleteProduct() |
| model | Controller | oneMinCommunity() |
| model | Controller | displayCommunities() |
| model | Controller | intToCommunity(intCommunity : int) |
| model | Controller | oneMinProduct(community : Community) |
| model | Community | oneMinProduct() |
| model | Controller | displayProducts(community : Community) |
| model | Community | displayProducts() |
| model | Product | getName() |
| model | Controller | deleteProduct(intCommunity : int, intProduct : int) |
| model | Community | deleteProduct(intProduct : int) |
| Req#5: Add species to a place | ui | Main | addSpecies() |
| model | Controller | oneMinPlace() |
| model | Controller | displayPlaces() |
| model | Place | getName() |
| model | Controller | displaySpeciesTypes() |
| model | Community | getSpeciesTypes() |
| model | Place | getSpeciesTypes() |
| model | SpeciesType | getSpeciesTypes() |
| model | SpeciesType | SpeciesType(description: String) |
| model | SpeciesType | getDescription() |
| model | Controller | addSpecies(intPlace: int, speciesName : String, photo : String, localPop : int, intSpeciesType : int) |
| model | Controller | intToPlace(intPlace : int) |
| model | Place | addSpecies(name: String, photo : String, localPop: int, intSpeciesType : int) |
| model | Place | availableSpecies() |
| model | Place | searchSpecies(speciesName: String) |
| model | Species | getName() |
| model | Species | Species(name : String, photo : String, localPop: int, intSpeciesType : int) |
| model | SpeciesType | intToSpeciesType(intType : int) |
| Req#6: Modify a species data in a place | ui | Main | modifySpecies() |
| model | Controller | oneMinPlace() |
| model | Controller | displayPlaces() |
| model | Place | getName() |
| model | Controller | oneMinSpecies(place : Place) |
| model | Controller | intToPlace(intPlace : int) |
| model | Place | oneMinSpecies() |
| model | Controller | displaySpecies(place : Place) |
| model | Place | displaySpecies() |
| model | Species | getName() |
| model | Controller | modifySpecies(intPlace : int, intSpecies : int, newName : String, photo : String, localPop : int, intNewSpeciesType : int) |
| model | Controller | searchSpecies(name: String):boolean |
| model | Place | searchSpecies(speciesName: String) |
| model | Species | getName() |
| Model | Controller | intToPlace(intPlace : int) |
| model | Place | modifySpecies(intSpecies : int, newName : String, photo : String, localPop : int, intNewSpeciesType : int) |
| model | Place | intToSpecies(intSpecies : int) |
| model | Species | setName(newName : String) |
| model | Species | setPhoto(newPhoto : String) |
| model | Species | setLocalPop(newLocalPop : int) |
| model | Species | intToSpeciesType(intType : int) |
| model | SpeciesType | intToSpeciesType(intType : int) |
| model | Species | setType(newSpeciesType : SpeciesType) |
| Req#7: Access a place information. | ui | Main | accessPlace() |
| model | Controller | oneMinPlace() |
| model | Controller | displayPlaces() |
| model | Place | getName() |
| model | Controller | accessPlace(intPlace : int) |
| model | Controller | intToPlace(intPlace : int) |
| model | Place | toString() |
| model | PlaceType | getDescription() |
| model | Department | getDescription() |
| model | Community | getName() |
| model | Place | displaySpecies() |
| Req#8: Access the information of the communities in a department. | ui | Main | accessDeptCommunity() |
| model | Controller | oneMinPlace(); |
| model | Controller | displayDepartments(); |
| model | Place | getDepartments() |
| model | Department | getDepartments() |
| model | Department | Department(description: String) |
| model | Department | getDescription() |
| model | Controller | displayDeptCommunities(intDepartment : int) |
| model | Controller | deptPlaceList(intDepartment : int) |
| model | Place | intToDepartment(intDepartment : int) |
| model | Department | intToDepartment(intDepartment : int) |
| model | Place | getCaregivingCommunity() |
| model | Community | getName() |
| model | Controller | accessDeptCommunity(intDepartment : int, intDeptCommunity : int) |
| model | Community | toString() |
| Req#9: Access the information of communities that face an absence of hospitals and a lack of schools. | ui | Main | accessHospSchooCommunities() |
| model | Controller | accessHospSchooCommunities() |
| model | Community | getChallenges() |
| model | Community | toString() |
| Req#10: Display the name of the place with the most species diversity. | ui | Main | mostDiverse() |
| model | Controller | oneMinPlace(); |
| model | Controller | mostDiverse() |
| model | Place | speciesCount() |
| model | Place | getName() |
| Req#11: Display the three places with the biggest area. | ui | Main | biggestArea() |
| model | Controller | biggestArea() |
| model | Place | getArea() |
| model | Place | getName() |