**First Part (Completed Projects):**

CompletedProjectsCollection = [

ProjectEntity: {

users: [{ // Related to ‘UsersCollection’

userNameLink: Number,

required: true

}],

technologies: [{ // Related to ‘TechnologiesCollection’

techNameLink: Number,

required: true

}],

projectName: {type: String, required: true} // Unique

descriptions: [{ // Free Text

date: date,

descrText: String

}],

screenshots: [{

internal: Boolean, //Internal: true, External: false

linkToSource: String, // if ‘internal’ == false

shot: Buffer,

}],

begin-end-Time: {

begin: DateTime,

end: DateTime

},

tags: [{ // Related to ‘TagsCollection’

fromCollection: Boolean, // ‘true’ if selected from ‘TagsCollection’

tagNameLink: Number, // if ‘fromCollection’ is ‘true’

tagName: String

}],

stage: { // Related to ‘stagesCollection’

stageNameLink: Number

}

estimation: [{

value: Number,

date: Date,

description: String

}]

}

]

**Second Part (Requested Projects):**

RequestedProjectCollection = [

RequestedProjectEntity: {

projectName: String,

descriptions: [{

date: Date,

descrText: String,

images: [{

img: Buffer

}]

attachments: [{

name: String,

date: Date,

att: Buffer

}]

}],

tags: [{ // Related to ‘TagsCollection’

fromCollection: Boolean, // ‘true’ if selected from TagsCollection

tagNameLink: Number, // if ‘fromCollection’ is ‘true’

tagName: String

}],

technologies: [{ // Related to ‘TechnologiesCollection’

techNameLink: Number

}],

conditions: { // Related to ‘ConditionCollection’

conditionNameLink: Number

},

screenshots: [{

internal: Boolean, //Internal: true, External: false

linkToSource: String, // if ‘internal’ == false

shot: Buffer,

}],

features: [Feature]

}

}]

**Auxiliary Collections:**

UsersCollection = [{

\_id: ObjectId,

login: String,

userName: String,

userSurname: String,

avatar: Buffer,

authHash: String

}]

TechnologiesCollection = [{

\_id: ObjectId,

techName: String,

techAvatar: Buffer

}]

TagsCollection = [{

\_id: ObjectId,

tagName: String,

tagLinks: [String]

}]

StagesCollection = [{

\_id: ObjectId,

stageName: String,

comissioned: DateTime,

decomissioned: DateTime

}]

ConditionCollection = [{ // ‘In Progress’ / ‘Estimated’ / ‘Discussed’

\_id: ObjectId,

conditionName: String,

conditionDescription: String,

comissioned: DateTime,

decomissioned: DateTime

}]

var Section = new Schema({ // ‘In Progress’ / ‘Estimated’ / ‘Discussed’

\_id: ObjectId,

sectionName: String,

sectionDescription: String,

opened: {type: Date, required true, default: Date.now}

closed: Date,

features: [Schema.Feature.\_id]

}

})

var Feature = new Schema({

\_id: ObjectId,

featureName: String,

featureOrder: String,

isNecessary: Boolean, // 'true' == Necessary, 'false' == Desirable

featureDescription: {

images: [{shortName: String, binBody: Buffer}],

extImagesLinks: [String],

attachments: [{fileName: String, binBody: Buffer}],

extLinks: [String],

lists: [[String]]

},

created: {type: Date, default: Date.now},

isImplemented: Boolean,

childFeatures: [SubFeature]

})

var SubFeature = new Schema({

\_id: ObjectId,

subFeatureName: String,

subFeatureOrder: String,

isNecessary: Boolean, // 'true' == Necessary, 'false' == Desirable

subFeatureDescription: {

images: [{shortName: String, binBody: Buffer}],

extImagesLinks: [String],

attachments: [{fileName: String, binBody: Buffer}],

extLinks: [String],

lists: [[String]]

},

created: {type: Date, default: Date.now},

isImplemented: Boolean

})