Will Cowan
EasyManage
Database Design
Week 3

Overview:

For EasyManage I am choosing to use MongoDB as my database. I am choosing a document database design for multiple reasons. The first is flexibility; from personal experience award reporting requirements are constantly changing and exceptionally variable among sponsors. Because of this I think using a document model is a better option. Document models allow for additional records to be made to a select group of records without having to make additions to the whole. My department attempted to use a relational database to keep track of award reporting information. It ended up being a big mess; we had hundreds of related tables and it was very difficult to keep the database normalized. We also struggled with database performance; complicated queries would often freeze the database. I believe all of the award reporting data can be handled in a single JSON document. I also believe that changes to award reporting data will not cause cascading updates. For example, changing one award's reporting requirement has no impact on the other awards stored in the database. There are also real-world limitations to one-to-many relationships. For example, one professor can only handle so many awards at a time due to time and facility restraints.

Data Models:

Award:

```
{
        "id": Object ID
        "Award_Number": Integer,
        "Award_Name": String,
        "Award_Sponsor": String,
        "Award_PI":String,
        "Dept_Number": Object, * Embedded Data from Department Document
        "Start_Date": String,
        "End_Date": Date,
        "Reporting_Period": String,
        "Special_Report":String,
        "Report_Status": String
}
```

User:
{
	“id”: Object ID
	“Employee_Number”: String,
	“First_Name”: String,
	“Last_Name”: String,
	“Email”:String,
	“Password”: String
	“Role”: String
}

Department:
{
	“id”: Object ID
	“Dept_Number”: Integer,
	“Dept_Name”: String
}

Comments:
{
	“id”: Object ID
	“Employee_Number”: Object *Embedded Object from User Document
	“Award_Number”: Object * Embedded from Award Document
	“Comments”: “String”
}

Notifications: - This is my stretch feature
{
	“id”: Object ID
	“Employee_Number”: Object *Embedded Object from User Document
	“Award_Number”: Object * Embedded from Award Document
	“Award_Status”: Object * Embedded from Award Document
}

PLEASE NOTE THAT I PLAN ON HASHING AND SALTING USER INFORMATION SUCH AS PASSWORDS FOR DATABASE SECURITY

Explanation for Data Models:

Department:
The purpose of this document is to store department identifying information. The purpose of this table is to make for quicker querying of data. Departments can have multiple awards assigned to them. By embedding the department number in the Award document, it will be much faster to query award info by department. I envision this being a fairly common request and wanted to make it as efficient as possible. For example, Adam Admin is very likely to want a list of all awards assigned to his department. His department head could want the information for various purposes such as facility planning or staffing decisions.

Data Example:
{
"id": 1
"Dept_Number": 1000,
"Dept_Name": "Biomedical Sciences"
}

User:
The purpose of this document is to store user log in information. All users will have to use it to log into the application. For example, Mary Manager will log in on a daily basis to pull information out of the application. I have added the "role" field. It is used to track a user's role in order to link notifications to a user's role. I have added an Employee Number field to track an employee's number, it is used to link Employees to report status change notifications. For example, once Alice Analyst completes a review, she will change the report status to complete ready for review and Mary Manager will receive an update.

Data Example:
{
"id": 1
"Employee_Number": 2000
"First_Name": "Mary",
"Last_Name": Manager,
"Email": "Mary@company.com",
"Password": "password"
"Role": "Manager"
}

Award:
The purpose of this document is to store all of the relevant award reporting information. It will form the basis for most of the application's data and queries. The Report Status field is a new field. It will be used to track the reports status. User's will get a notification that an award's status has changed based on their role. For example, Mary Manager might want to see all the reporting information for an award. She would query information from this document. When Mary approves a report, Alice Analyst will get a push notification that the report has been signed and approved.

Data Example:

```
{
        "id": 1,
        "Award_Number": 1500,
        "Award_Name": "Biomedical Sciences Research",
        "Award_Sponsor": "NIH",
        "Award_PI": "Dr. John Doe",
        "Dept_Number": 1000,
        "Start_Date": "10/1/21",
        "End_Date": "10/31/22,
        "Reporting_Period": "Annual",
        "Special_Report": "Effort Report"
        "Report_Status": "Incomplete"
}
```

Comments:
This document serves to enhance work flows by allowing employees to leave comments regarding their work for other users. For example, if Alice Analyst was working on a report but needed more information, she could leave a comment on the award that said "Waiting on department to clarify salary expenses". This will leave a trail that is easily accessible to anyone working on the award. That way the number of emails between employees are reduced.

Data Example:

```
{
        "id": 1
        "Employee_Number": 2000
        "Award_Number": 1500
        "Comments": "Report is ready, please submit"
}
```

Notifications:
This document serves to provide employees with real time notifications of award status changes based on their role. For example, when Alice Analyst changes an award's report status from WIP to Complete Ready for Review Mary Manager will get a notification. I believe there is a way to easy set up push notification in Ionic; however, I want there to be a historical record that appears on the users home page. This will be more difficult to code and is my stretch feature.

Data Example:

{
	"id": 1
	"Employee_Number": 2000
	"Award_Number": 1500
	"Report_Status": "Complete Ready for Review
}