Salesforce Bootcamp - Final Project

Objective

Put into practise the concepts learnt during bootcamp. You will have to:

- Understand the proposed business reality
- Create and apply a data model in Salesforce
- Use apex to retrieve and store data
- Create unit test methods to cover code
- Create tabs, a simple lightning component and a Lightning App

Your work here will put into perspective the way you handle obstacles and structure your code.

Prerequisites

- Bootcamp training
- Laptop with sfdx environment configured

Business overview

Universal Containers Lawyers is a company that provides legal advisory to their customers. They consume data from an external system that stores the information into Heroku on a daily basis.

They want to adopt Salesforce and need to migrate the information stored in Heroku into an org. The data is retrieved as a JSON, and contains two types of entities: ‘Legal Advisor’, and ‘Client’. There is a web service in Heroku that contains the json file. Also, given that this information changes periodically, they need a way to do this process automatically every day.

Legal Advisors are firms that provide legal services to different clients. Clients are the ones that hire the advisors services.

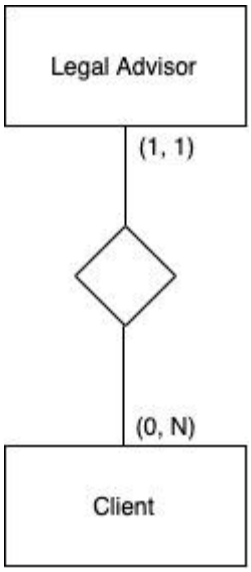
Legal Advisors are uniquely identified by its Account Number.

Clients are uniquely identified by its Account Number + Client Number. Note that a client can only be related to just one Legal Advisor. Each Legal Advisor assigns a unique Client Number. However, the Client Number may not be unique across Legal Advisors.

They ask that all classes you create have the prefix ‘ucl_’. For example: ‘ucl_LegalAdvisors.cls’

It is worth noting that Universal Containers has a QA team that will be in charge of validating the functionality you implement.

Proposed Data Model



Legal Advisor

Field name	Field type
Account Number	Text
Account Name	Text
Account Status	Text
As of Date	Date

Client

Field name	Field type
First Name	Text
Last Name	Text
Email	Text
Phone	Text
Address	Text
Client Number	Text

Json sample data


```
[{
  "LegalAccounts": [
    {
      "AccountNumber":"9afccdd3-e7c4-4789-a465-1734886633d2",
      "AccountName":"Centidel",
      "AccountStatus":"Enabled",
      "AsOfDate":"2/13/2019",
      "Clients":[
        {
          "FirstName":"Foster",
          "LastName":"Burn",
          "Email":"fburn0@nydailynews.com",
          "Phone":"425-154-5410",
          "Address":"49018 Bashford Road",
          "ClientNumber":"b4110a35-7ca3-4a53-8d94-ea78c5489052"
        },
        {
          "FirstName":"Minetta",
          "LastName":"Pol",
          "Email":"mpol1@thetimes.co.uk",
          "Phone":"704-528-1659",
          "Address":"21 Lukken Drive",
          "ClientNumber":"c0c93578-6bbf-4bcc-a786-4d1e45ea0c96"
        }
      ]
    }
  ]
}]
```

Heroku information

URL: https://altimetrik-bootcamp.herokuapp.com/
Resource: /LegalAccounts

Identified tasks

- Create Kanban board in Trello to track stories (Create > Start with a template > Kanban Template). Cards need to be created based on the requirements and information of this document and need to have all the information needed in the description field to achieve the task (NO one line description like “create X object”). Think about someone else reading just the card and they will have to know what exactly needs to be done. Name the board (like you did for the trailmix) with your name “ - Project” (ex: “Juan Bessonart - Project”), share the board with Sebastian and Juan.
- Create a branch with the name ‘stage’ and your initials (ex: ‘stage-mc’) and a branch ‘dev’ and your initials (ex: ‘dev-mc’) in the Github repository that you already created as part of the Trailmix. Use the ‘dev’ branch as your working branch.
- Implement the suggested Salesforce data model. *Create more fields or objects if you think it's justifiable.*
- Provide a nightly service that can update/insert data from the Heroku service into Salesforce. Don’t forget to use the prefix ‘ucl_’
- Legal Advisors with an ’Enabled’ status should be inserted. Advisors with a ‘Disabled’ status should not be processed, unless the record already existed in Salesforce with a status of ’Enabled’ in which case it should be updated to reflect the new reality.
- If a Legal Advisor record already exists, the record and its related customers should be still updated.
- If a Client that was inserted previously by the service is no longer coming in the JSON file, it has to be deleted from the Org.
- Universal Containers wants to know how many clients an advisor has in Salesforce. Find the best way to display this information.
- If a Legal Advisor is deleted by an admin, there is no reason to keep their clients.
- If an error occurs during the process you implemented, an error has to be logged, and an email has to be sent to the System administrator. The errors need to be logged somewhere with easy access to the business team. They have to be kept in the system for at least a month.
- The application will be deployed to a production environment, so 75% of code coverage is required (as close as 100% would be better).

- Create custom object tabs for Client and Legal Advisor so an admin can create records using the objects' record pages. Make sure that the Legal advisor has a Related List of Clients in its page, and all custom fields display correctly in the page.
- Admins might add Clients directly in Salesforce using the Client record page. These records should not be deleted by the nightly service you implement. You must provide a way to avoid this.
- Create a small lightning component that only has a button that invokes the service you implemented directly. It should display a simple toast if the process was called with the result information.
- Create a lightning tab based on this lightning component.
- Lastly, create a new Lightning app that contains the tabs mentioned in the previous 2 steps. It should be called 'Universal Containers Lawyers app'. Also, add a way for the user to easily view the error logs.

Suggested steps

- Create a developer org (scratch orgs are admitted)
- Create tickets/cards in Trello.
- Create a 'dev' and 'stage' branch in your repository.
- Implement the suggested data model in your org.
- Implement all the classes and code you think is necessary. Also, implement a schedulable class that executes your integration every night (a batch is not needed for this scenario)
- Create all unit tests for maximum coverage possible
- Create custom tabs for the created objects and the lightning application mentioned
- When you consider that your work is done, create a MR to 'stage' branch and share your credentials with your reviewer

Tips

- Upload the code to your dev branch periodically
- Using external Ids might help you to upsert data.
- Assume that there won't be a large number of records. Legal advisors won't surpass the 100 records and the Clients won't be above 500 records in total.

Useful links

Invoking Callouts Using Apex
Apex REST callouts unit
Schedule Jobs Using the Apex Scheduler
Adding a related list

[Trello Kanban Board]