Project Proposal

Task Import Tool – HubSpot CRM

Name, student number

Higher Diploma in Science in Computing

Software Development

Date

Background

HubSpot is an all in one customer relationship management (CRM) platform with over 95,000 customers that allows users to organise, track and build better relationships with their leads and existing customers. This means that they use this platform as a database where they store all their contacts, companies, deals and support tickets. They also utilise it to host their websites, create marketing campaigns and create other content to generate leads and nurture their existing leads and customer base. At present, users can import objects like contacts, companies, deals, tickets and even custom objects but engagements such as meetings, emails, calls and tasks cannot be imported via a .CSV or a .XLSX file. This can cause some friction, for new users especially. They are migrating to a new tool, only to find out that they cannot import their engagements. The only way they can achieve this goal is by using HubSpot's API (application programming interfaces) to create and associate these engagements with their objects, or use a paid third-party tool that will do this for them. To make migrating to this platform easier and allow for a smoother transition, where no tasks will be missed, my project will focus on creating a web application that can be integrated with a HubSpot account, using OAuth2. As defined on HubSpot's developer website, "OAuth is a secure means of authentication that uses authorization tokens rather than a password to connect your app to a user account" (HubSpot.com, 2021).

It is important to clarify that in this context, a portal refers to a HubSpot user's account. It can also be referred to as their CRM.

Objectives

The objective of this project is to create a web application that will allow users to import tasks to their HubSpot portal, something that is not currently possible using a .CSV or a .XLSX file. The users will be able to access the web application in their browser, which will bring them to HubSpot's OAuth 2.0 Server. Here, the user will need to review the requested permissions and grant the application access. Once they have granted access, they will be redirected back to the application with an authorization code in the query string. The application sends a request to the OAuth 2.0 server to exchange the authorization code for an access token. This is what will be used to make requests using HubSpot's API for that specific portal. Every user that uses HubSpot's UI would have their own access token.

Users will be able to import their tasks via this application, using a form where they can upload a .CSV file. Node.js will be used to parse through that and call the HubSpot API. The output will be the tasks (engagements) arriving in HubSpot.

Technical Approach

In order to become familiar with the language and technologies needed to create this web application, the completion of "The Complete Node.js Developer Course (3rd edition)" Udemy course will be executed. This will need to take place before the design stage of this web application can begin. In addition to this, *Write Modern Web Apps with the MEAN STACK* by Jeff Dickey will be used as a resource throughout this project. More specific to using HubSpot's APIs, there are several resources aimed specifically at developers interested in using their APIs so these will be consulted throughout the project.

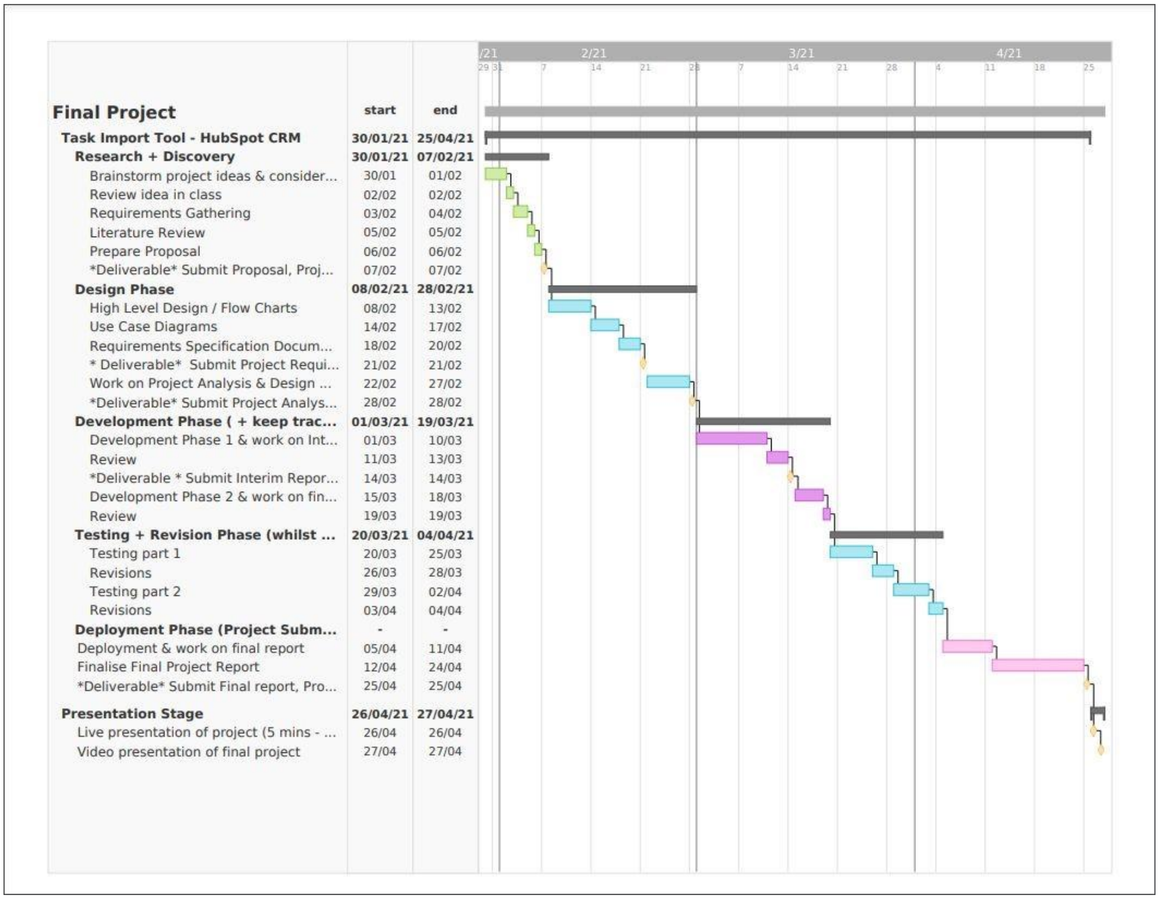
Thorough research on how to integrate an application with a HubSpot Portal will need to be carried out as well. A few different HubSpot APIs will be used to be able to achieve the objective of this application. For example, the "Search for contacts by email, name, phone number, or company" endpoint. Since it is possible that the contacts that these tasks will be associated with, don't already exist as contacts, they will not have a Customer ID, which is the unique identifier needed to be able to associate an engagement to an object. This means that users will more than likely be importing these tasks and associating them with an email address, rather than with a Customer ID. Therefore by using the aforementioned endpoint, the application will be able to search the user’s portal and if a contact already exists with that email address, return the visitor ID (vid) of that contact. This can then be used to associate the contact to the task. If no contact exists with that email address, a new contact can be created with the use of the “Create contact” endpoint. Once the contact is created, a vid will be generated, which can then be used to associate the newly created contact to the task. Tasks will be created in the user’s portal using HubSpot’s “Create an engagement” endpoint.

Visual studio code will be used for the development of this application. A HubSpot developer and test account will also be needed to test the application. The developer account is where the application will be listed, whereas the test account will allow the testing of APIs and the application without impacting data from a real HubSpot account.

Special resources required

HubSpot’s API documentation which can be found on their developer website.

Project Plan



Final Project	start	end
Task Import Tool - HubSpot CRM	**30/01/21**	**25/04/21**
Research + Discovery	**30/01/21**	**07/02/21**
Brainstorm project ideas & consider...	30/01	01/02
Review idea in class	02/02	02/02
Requirements Gathering	03/02	04/02
Literature Review	05/02	05/02
Prepare Proposal	06/02	06/02
Deliverable Submit Proposal, Proj...	07/02	07/02
Design Phase	**08/02/21**	**28/02/21**
High Level Design / Flow Charts	08/02	13/02
Use Case Diagrams	14/02	17/02
Requirements Specification Docum...	18/02	20/02
* Deliverable* Submit Project Requi...	21/02	21/02
Work on Project Analysis & Design ...	22/02	27/02
Deliverable Submit Project Analys...	28/02	28/02
Development Phase (+ keep trac...	**01/03/21**	**19/03/21**
Development Phase 1 & work on Int...	01/03	10/03
Review	11/03	13/03
*Deliverable * Submit Interim Repor...	14/03	14/03
Development Phase 2 & work on fin...	15/03	18/03
Review	19/03	19/03
Testing + Revision Phase (whilst ...	**20/03/21**	**04/04/21**
Testing part 1	20/03	25/03
Revisions	26/03	28/03
Testing part 2	29/03	02/04
Revisions	03/04	04/04
Deployment Phase (Project Subm...	-	-
Deployment & work on final report	05/04	11/04
Finalise Final Project Report	12/04	24/04
Deliverable Submit Final report, Pro...	25/04	25/04
Presentation Stage	**26/04/21**	**27/04/21**
Live presentation of project (5 mins - ...	26/04	26/04
Video presentation of final project	27/04	27/04

* PDF version attached for better visibility of Gantt Chart

Technical Details

The MEAN (MongoDB, Express.js, AngularJS, and Node.js) Stack will be used to build, test and deploy this web application. The MEAN Stack will allow the development of this application using only one language, JavaScript, for both client and server side since MongoDB, Express.js, AngularJS and Node.js are all JavaScript technologies. Below is a brief description of these technologies that will be used in this project:

- MongoDB will be used as the database system as it can be used by server side applications to store data as JavaScript Object Notation JSON documents (IBM, 2021).
- Express - According to IBM (2019) Express is a backend (server side) web application framework that deals with all the interactions between the front-end and the database and it uses Node.js.
- AngularJS is a client side (front-end) framework (IBM, 2021). Dickey (2014) suggests that it provides many helpful tools such as DOM manipulation, while providing a modular structure thus being easily testable.
- Node.js is a backend (server side) runtime environment that is based on Chrome’s V8 Javascript engine. One of the advantages of using Node.js is that it is highly scalable (IBM, 2019).

One of the main reasons the MEAN stack was chosen for this project is because all of the technologies in the MEAN stack are free and open-source, meaning that the application can be built free of charge. In addition to this, it uses JSON as the data interchange format on all the layers which will conveniently allow working with HubSpot’s APIs.

Evaluation

To evaluate whether the application is working as it should, several software tests will be carried out. Running these tests will prevent bugs and improve the overall performance of the application as any flaws can be detected before the launch of the application. First, unit testing will be performed, which ensures that each individual software unit works as intended. Once unit testing along with any necessary amendments have been completed, integration testing will be carried out to verify whether all the different components such as the server side, client side and database function well together. Moving on to system testing, a usability test will be performed. To run this test, a .CSV file will be generated with fake, randomised information that can be used for testing purposes. As all HubSpot employees are provided with their own test portal, this test will be carried out in such a test portal, meaning the data in a real user’s portal will not be affected as a result of this test. Lastly, once any revisions have been made based on the results of the usability testing, acceptance testing will be carried out using a test HubSpot portal.

Signature

____

Date

Bibliography:

Developers.hubspot.com. 2021. How to use HubSpot's APIs. [online] Available at: <https://developers.hubspot.com/docs/api/developer-guides-resources> [Accessed 6 February 2021].

Developers.hubspot.com. 2021. Working with OAuth. [online] Available at: <https://developers.hubspot.com/docs/api/working-with-oauth> [Accessed 6 February 2021].

Dickey, J., 2014. Write modern web apps with the MEAN stack. San Francisco, CA: Peachpit Press.

Guru99.com. 2021. What is System Testing? Types & Definition with Example. [online] Available at: <https://www.guru99.com/system-testing.html> [Accessed 7 February 2021].

Ibm.com. 2019. mean-stack-explained. [online] Available at: <https://www.ibm.com/cloud/learn/mean-stack-explained#toc-components-oVDOIhYC> [Accessed 6 February 2021].

Ibm.com. 2021. mongodb. [online] Available at: <https://www.ibm.com/cloud/learn/mongodb#toc-use-cases-_yepkehV> [Accessed 6 February 2021].

Ibm.com. 2021. What is software testing?. [online] Available at: <https://www.ibm.com/topics/software-testing> [Accessed 7 February 2021].