Austin Dunn

VBA Semester Project

Executive Summary:

The Center for Language Studies (CLS) at Brigham Young University offers courses in over 40 different languages. In addition to offering language courses, the CLS serves other language departments in the College of Humanities by administering a capstone exam to language majors before they exit the program. This exam is called the Oral Proficiency Interview (OPI). Students sign up for two dates and times when they are available to take the OPI and receive a test schedule during those requested times.

The OPI is a phone interview with a certified tester and measures the speaking proficiency of students. Typically, the interview takes 30 minutes to complete. In a normal semester, the CLS will test roughly 400 students. The student administrators at the CLS spend a lot of time retrieving test requests from a Google Doc, entering the requests into a third-party website, and notifying students about their scheduled test times. Additionally, the student administrators send a reminder email to all students who will take the OPI the following day. This is to help students avoid paying a $55 no-show fee if they arrive late or don't arrive at all.

The system built for this project automates the majority of the test arranging process so that student administrators have more time to build various reports and serve the adjunct faculty for the CLS. The system performs the following: (1) retrieves sign up information from the Google Doc, (2) requests all new tests on the third-party website, (3) sends a reminder email to students who will be taking the test the next day, and (4) stores the newest test requests in a database. The student administrators have the option of performing all or only some of these functions through the use of various buttons.

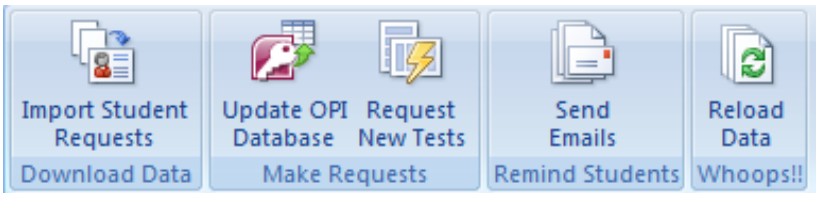
Implementation Documentation:

In order to consider my project successful, the minimum functionality necessary includes the following:

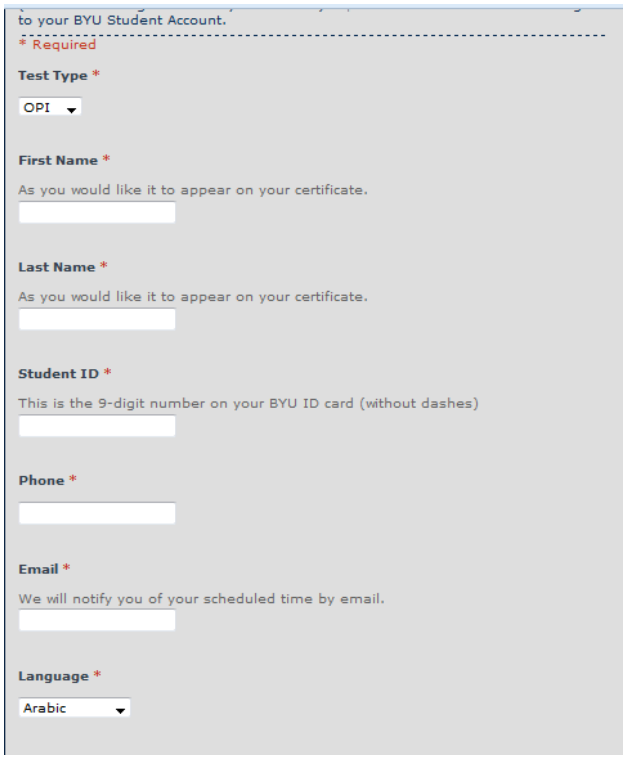
- Part 1: Retrieve sign up information. Students sign up using a Google Form that is linked to a Google Spreadsheet (titled: VBA OPI Requests)
- Part 2: Insert all new requests into an Access database. The database will be used for a different personal project, but I designed the insert function for this project.
- Part 3: Determine which entries on the spreadsheet are new entries and submit any new test requests on the Language Testing International website (languagetesting.com/client).
- Part 4: Update the Google Spreadsheet to record that the test has been requested. This ensures that I don't request a test for the same student over and over.
- Part 5: Look through the spreadsheet and find all tests that are scheduled for the following day. Email a reminder message to all students whose test is the next day, thus helping them avoid paying a $55 no-show fee.

I created a new tab on the Excel ribbon for my project. It has buttons that activate each part of my project.

Austin Dunn



Part 1: For this part of my project, I designed a Google Form and Spreadsheet where students sign up to take the test. I send a link to the form to all students who are language majors taking their capstone course in the current semester. All students in a capstone language course take the Oral Proficiency Interview (OPI). To sign up for the OPI, the students fill out the form and responses are stored in the spreadsheet. Click here for a link to the full form or you can see a snapshot of what the form looks like below:

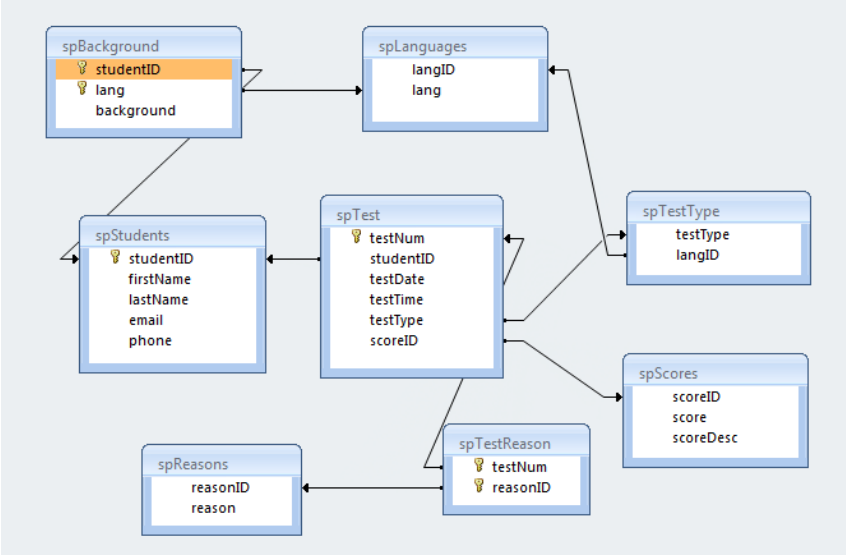


The Spreadsheet that stores student responses is published to the web so that I can import the data from the Google Spreadsheet into Excel. I want to note that published does not mean that it is public. Only someone with the link can access the information. To import the page, I used the Agent Class that Professor Allen gave us. I used the Agent to login to my Gmail account, go to the Spreadsheet URL, and import the page. Then I remove all the extra rows, columns, and data that I do not need to submit the request. I place all necessary data for Part 2 into a tab called "OPIs". All this code is run when the user clicks the 'Import Student Requests' button on the 'OPI Tools' tab. Here is a snapshot of what the data looks like once it has been imported from the Google Spreadsheet:

Emailed	Timestamp	Test Type	First Name	Last Name	Student ID	Phone	Email	Language	Date Choice 1	Time Start	Time End	Date Choice 2	Time Start	Time End	Reason	How did you learn	Have you ha	Ordered	Test
	4/1/13 1:37 PM	OPI	Luiz	Maykot	98704983	8014221201	cls@byu.edu	Portuguese	4/9/2013	10:00 AM	12:00 PM	4/11/2013	2:00 PM	5:00 PM	Port 491	Native	Yes	4/16/2013	
	4/9/13 1:37 PM	OPI	Austin	Dunn	654198312	8014221201	cls@byu.edu	Portuguese	4/19/2013	10:00 AM	12:00 PM	4/22/2013	1:30 PM	3:30 PM	Port 491	RM	No	4/16/2013	
	4/9/13 1:53 PM	OPI	Michele	Price	751421312	8014223477	michele_pric	Spanish	4/29/2013	2:00 PM	4:00 PM	4/24/2013	11:15 AM	2:00 PM	Fren 491	Heritage Speaker	No	4/16/2013	
	4/9/13 1:56 PM	OPI	Julia	Botita	841465495	8014223765	flsr-dept@by	German	5/2/2013	10:00 AM	2:00 PM	5/3/2013	10:00 AM	2:00 PM	Germ 400	Beginner	No		
	4/9/13 1:59 PM	OPI	Agnes	Welch	987464201	8014225199	agnes_welch	Mandarin	5/6/2013	10:30 AM	3:45 PM	5/8/2013	10:30 AM	3:45 PM	Chin 495	Native	Yes		
	4/9/13 2:02 PM	OPI	Robert	Erickson	135467092	8014225193	rob_erickson	French	5/10/2013	8:00 AM	5:00 PM	5/14/2013	10:00 AM	3:00 PM	Fren 491	Professional	No		
	4/9/13 2:35 PM	OPI	Ray	Clifford	559721135	8014223263	RayC@byu.ed	German	5/13/2013	1:00 PM	4:00 PM	5/14/2013	9:00 AM	1:00 PM	Germ 400	RM	No		
	4/9/13 2:37 PM	OPI	Michael	Bush	644462103	8014224515	michael_bus	French	5/1/2013	9:30 AM	12:00 PM	5/2/2013	1:00 PM	4:00 PM	Fren 491	RM	No		
	4/9/13 2:39 PM	OPI	Bob	Bockholt	778910642	8014226129	bob_bockhol	Portuguese	5/7/2013	10:45 AM	12:15 PM	5/9/2013	10:30 AM	12:30 PM	Port 491	Professional	No		

Note: For purposes of this project, I am only working with one type of test called the OPI. The Center for Language Studies administers various tests, but the project is only set up to request OPIs. The reason for this is that we perform nearly 1000 OPIs per year and only 100 or 200 WPTs. The sign up for WPT is not nearly as important, but may be a functionality I add in the future.

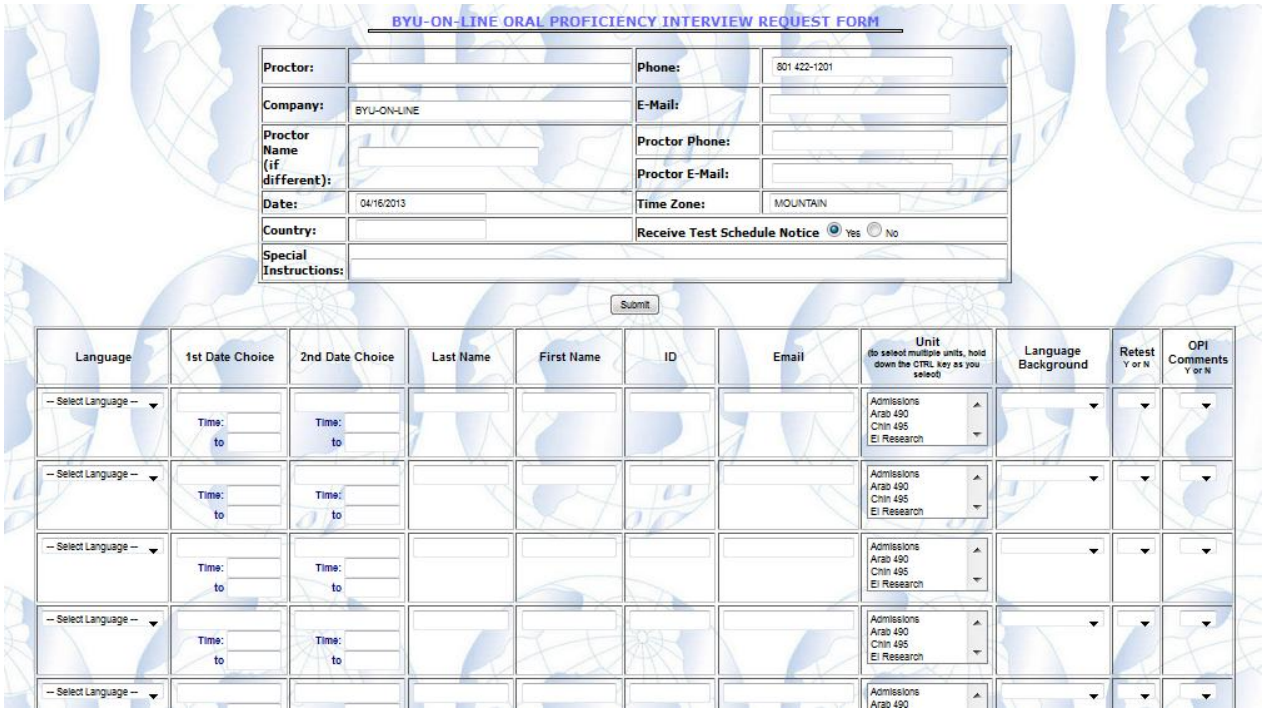
Part 2: I created an OPI database. The database will store test requests and student information. It eventually will store scores and other data about students and will be useful for reporting, but that is outside the scope of this project. However, for this project, I wanted to write the piece of the program that added new test requests to the database. This was the perfect place to add this 'insert' functionality because this project already imports new requests from my Google Doc. Now I simply had to create the insert piece. In my code, I had to ensure that any students whose names were already in the database were not added again, but a new 'test' instance could be added if a student takes tests in more than one language. I did this through the use of four 'insert into' statements, which update different tables. Here is the data structure from my database:



The Students, Background, Test, and TestReason tables are all updated with information that is imported into this project. The rest of the tables are static and do not change. To activate this portion of the project, the user clicks the 'Update OPI Database' button on the 'OPI Tools' tab.

Part 3: This part of the project requests the test on the Language Testing International (LTI) website. The code checks the column labeled "Ordered" for a date. If no date is found in the cell of the request row, it means that the test has not been requested before. The Agent class opens the LTI site in Internet Explorer. The site is secured so the Agent logs in. Once logged in, the agent finds the Test Requests page, which looks like this:

Austin Dunn



BYU-ON-LINE ORAL PROFICIENCY INTERVIEW REQUEST FORM

Proctor:		Phone:	801 422-1201
Company:	BYU-ON-LINE	E-Mail:	
Proctor Name (if different):		Proctor Phone:	
		Proctor E-Mail:	
Date:	04/16/2013	Time Zone:	MOUNTAIN
Country:		Receive Test Schedule Notice	Yes / No
Special Instructions:			

Submit

Language	1st Date Choice	2nd Date Choice	Last Name	First Name	ID	Email	Unit (to select multiple units, hold down the CTRL key as you select)	Language Background	Retest Y or N	OPI Comments Y or N
-- Select Language --	Time: to	Time: to					Admissions Arab 490 Chin 495 EI Research			
-- Select Language --	Time: to	Time: to					Admissions Arab 490 Chin 495 EI Research			
-- Select Language --	Time: to	Time: to					Admissions Arab 490 Chin 495 EI Research			
-- Select Language --	Time: to	Time: to					Admissions Arab 490 Chin 495 EI Research			
-- Select Language --							Admissions Arab 490			

There are only 10 request rows on this page. However, there may be more than ten students that sign up for the OPI on any given date. The project is designed to loop through the request info on the 'OPI' Excel sheet and add a request one by one. If 10 requests have already been entered and there are still more tests that haven't been requested, the form will submit and a new request page on the LTI site will open. The code then continues to add requests until all unrequested tests have been submitted on the LTI site.

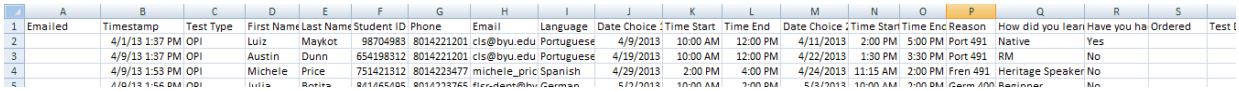
Here is a view of the form filled in:

Submit

Language	1st Date Choice	2nd Date Choice	Last Name	First Name	ID	Email	Unit (to select multiple units, hold down the CTRL key as you select)	Language Background	Retest Y or N	OPI Comments Y or N
Portuguese	4/9/2013 Time: 10:00 AM to 12:00 PM	4/11/2013 Time: 2:00 PM to 5:00 PM	Maykot	Luiz	98704983	cls@byu.edu	Korea 495 Lang. Cert. Misc Port 491	Native	Y	Y
Portuguese	4/19/2013 Time: 10:00 AM to 12:00 PM	4/22/2013 Time: 1:30 PM to 3:30 PM	Dunn	Austin	654198312	cls@byu.edu	Korea 495 Lang. Cert. Misc Port 491	RM	N	Y
Spanish	4/29/2013 Time: 2:00 PM to 4:00 PM	4/24/2013 Time: 11:15 AM to 2:00 PM	Price	Michele	751421312	michele_price@byu.edu	EI Research FLAS Post FLAS Pre Fren 491	Heritage Speaker	N	Y
German	5/2/2013 Time: 10:00 AM to 2:00 PM	5/3/2013 Time: 10:00 AM to 2:00 PM	Botita	Julia	841465495	flsr-dept@byu.edu	FLAS Post FLAS Pre Fren 491 Germ 400	Beginner	N	Y
Mandarin	5/6/2013 Time: 10:30 AM to 3:45 PM	5/8/2013 Time: 10:30 AM to 3:45 PM	Welch	Agnes	987464201	agnes_welch@byu.edu	Admissions Arab 490 Chin 495 EI Research	Native	Y	Y
French	5/10/2013 Time: 8:00 AM to 5:00 PM	5/14/2013 Time: 10:00 AM to 3:00 PM	Erickson	Robert	135467092	rob_erickson@byu.edu	EI Research FLAS Post FLAS Pre Fren 491	Professional	N	Y

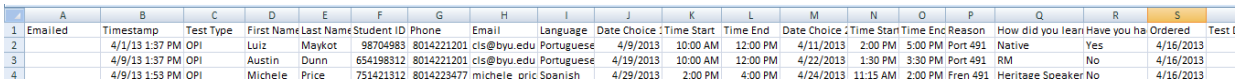
I need to ensure that a request for the same student is not submitted on this form multiple times. This would lead to a lot of confusion for the employees at LTI who schedule our tests for us. To make sure I avoid this, the code will enter the current date on the 'OPI' sheet next to each request after the data has been entered on the website. By the end, all requests should have a date next to them.

The following is what a request looks like when it has not been ordered:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Emailed	Timestamp	Test Type	First Name	Last Name	Student ID	Phone	Email	Language	Date Choice 1	Time Start	Time End	Date Choice 2	Time Start	Time End	Reason	How did you learn	Have you ha	Ordered
2		4/1/13 1:37 PM	OPI	Luiz	Maykot	98704983	8014221201	cls@byu.edu	Portuguese	4/9/2013	10:00 AM	12:00 PM	4/11/2013	2:00 PM	5:00 PM	Port 491	Native	Yes	
3		4/9/13 1:37 PM	OPI	Austin	Dunn	654198312	8014221201	cls@byu.edu	Portuguese	4/19/2013	10:00 AM	12:00 PM	4/22/2013	1:30 PM	3:30 PM	Port 491	RM	No	
4		4/9/13 1:53 PM	OPI	Michele	Price	751421312	8014223477	michele_pric	Spanish	4/29/2013	2:00 PM	4:00 PM	4/24/2013	11:15 AM	2:00 PM	Fren 491	Heritage Speaker	No	

The following is what a request looks like that has been ordered:



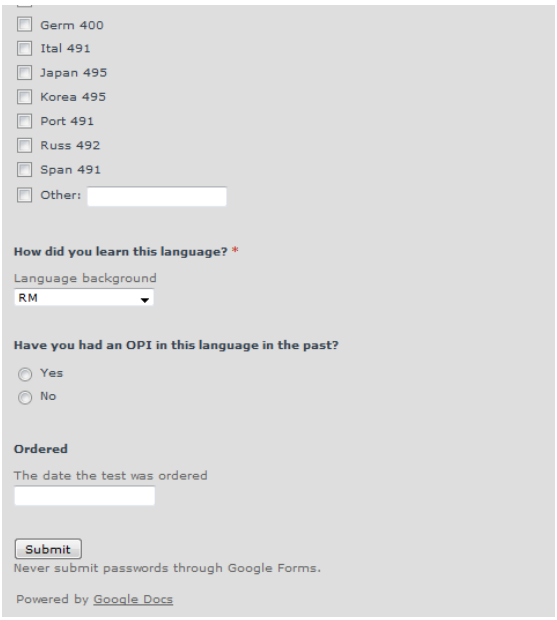
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Emailed	Timestamp	Test Type	First Name	Last Name	Student ID	Phone	Email	Language	Date Choice 1	Time Start	Time End	Date Choice 2	Time Start	Time End	Reason	How did you learn	Have you ha	Ordered
2		4/1/13 1:37 PM	OPI	Luiz	Maykot	98704983	8014221201	cls@byu.edu	Portuguese	4/9/2013	10:00 AM	12:00 PM	4/11/2013	2:00 PM	5:00 PM	Port 491	Native	Yes	4/16/2013
3		4/9/13 1:37 PM	OPI	Austin	Dunn	654198312	8014221201	cls@byu.edu	Portuguese	4/19/2013	10:00 AM	12:00 PM	4/22/2013	1:30 PM	3:30 PM	Port 491	RM	No	4/16/2013
4		4/9/13 1:53 PM	OPI	Michele	Price	751421312	8014223477	michele_pric	Spanish	4/29/2013	2:00 PM	4:00 PM	4/24/2013	11:15 AM	2:00 PM	Fren 491	Heritage Speaker	No	4/16/2013

Notice that the only difference is the presence of a date in the 'Ordered' column. The date will always be the date the test request was ordered.

Part 4: In order for this project to be fully effective, the ordered date entered in Part 3 must show up the next time I import the data from my 'VBA OPI Requests' spreadsheet. If the date does not show up the next time the data is imported, I would erroneously request the same tests over and over.

This solution requires that I change the Google Spreadsheet so that it includes the newly added date. That way, when I import the data the next day, the tests I requested will have an ordered date and will not be requested again. However, I was unable to change the original Google Spreadsheet using the Agent class. Spreadsheets on Google can only have one form and I am already using the Form for the sign up, so I couldn't add the date by using a form. To solve this problem, I created a mirrored Google Spreadsheet and Form called 'VBA OPI Orders'. It is an exact copy of the form students use to sign, except the Form for this new spreadsheet includes an additional field called "Ordered Date".

Here is a screenshot of the Order form:



Notice that this form has the addition of the 'Ordered' date field. Here is a link to the complete form.

When I request each test on the LTI site, the code will open the Google Form attached to VBA OPI Orders and it will make an entry with the request data and the request date. Then, the next time I import the data, the code imports the VBA OPI Requests spreadsheet and the VBA OPI Orders Spreadsheet. OPI Requests are imported and formatted as described in Part 1. However, I also import the data from OPI Orders and the code pastes all the 'Ordered Dates' into my 'OPI Sheet'.

This solution was an effective work around. The new Google Spreadsheet, OPI Orders, will have the same number of rows as the OPI Requests form because each student submission on OPI Requests is entered into the form for OPI Orders when I request the test. Because both Spreadsheets contain the same number of rows, I can be sure that the date on the Orders sheet matches the student name on the Requests sheet.

Note: Part 3 & 4 are both activated by the same button – 'Request New Tests'

Part 5: Finally, I hoped to save a lot of time for the student administrators by automating the reminder emails they send to students. In the past, students often forgot to show up on time or show up at all for their OPI appointments. BYU is charged a fee by Language Testing International if students do not keep their appointments. Students in turn are told to pay the fee for not showing up. To help students remember to show up and avoid the no-show fee, this project sends a reminder email to the student the day before their OPI is scheduled. The email function is activated by the button labeled 'Send Emails'. The project notifies me via text that it has finished once all the emails have been sent.

The student administrators update the Google Spreadsheet with the test dates of each student. Since that data is already being downloaded, it is fairly simple to create an email that can be sent to students. The text of the email is below:

```
Dear <NAME>,

This is a reminder that your OPI is tomorrow.  Please refer to the information below for details.

Your OPI has been scheduled for <DATE&TIME>. Please arrive in the testing office 1141 JFSB 5 to 10
minutes early. The office (Humanities Learning Resource) is located on the first floor of the JFSB in
the north hallway.  Remember that you can request the cancelation of your test up to 24 hours before
this scheduled time.  But, if you don't cancel and fail to show up or show up late for the test, you
will be charged a no-show fee of US$ 55.

As always, please remember that we enforce the BYU Honor Code and its Dress and Grooming Standards.
And don't forget to bring your BYU ID.

Best,

Center for Language Studies
(801) 422-1201
```

Once all new tests have been requested, the student administrator sends the reminder emails. By clicking the 'Send Emails' button on the 'OPI Tools' tab, the project finds all students whose test will be the following day and sends the email above. The project replaces the <NAME> and <DATE&TIME> fields with the appropriate student name and test date.

There is an exception in the project that looks for a test date of Monday if the current date is Friday. This allows the students testing on Monday of the following week to still receive a reminder email. The text of the reminder email is changed to state that the test is on Monday instead of saying 'your OPI is tomorrow'.

Discussion of learning and conceptual difficulties encountered:

In performing this project, I had in mind the goal that I wanted to learn how to do the following in VBA: (1) interface with a website, (2) automate sending emails, and (3) update and query information from a database. We covered all of these topics in class, but I didn't feel like I had a strong enough understanding to be able to do them on my own. Applying them in a personal project was very helpful in my goal. I had some difficulty understanding how to use the source code from a website to find the right names for input boxes and submit buttons. Reviewing the assignment from class helped me figure that out, and I used the internet as well.

Another difficulty I had was opening a connection to an Access database. The code from our class discussion did not work for me at first and it took me some time to figure out the reason. I had originally set up my database tables in MySQL Workbench and I linked the tables in Access rather than importing them. Apparently, VBA cannot make a connection to tables that are linked instead of imported (which Microsoft said is a known problem). To get around this problem I simply imported the tables to Access instead of linking them. Then the connection worked just fine.

One concept I was really hoping to learn was how to send updates to a Google Spreadsheet. My project required that I download information from the Google Spreadsheet, make a change, and

then update the Google Spreadsheet. The first time I imported the spreadsheet to excel, however, nearly all of the information was missing. I discovered that I could work around it by publishing the spreadsheet to the web. This solved my problem of downloading the data. I could not find a way to update the same spreadsheet, though. Professor Allen mentioned a concept I was not familiar with called Google API's. I did some research on this but concluded I didn't have enough time to write an API for my project. Professor Allen also suggested I could create a second Google Spreadsheet that stored the information I wanted to upload; then I could import both spreadsheets and have all the data. This worked very well. I would have liked to have gotten the API to work, and I even took a stab at it; however, given my time constraints it simply wasn't doable. I will probably learn it anyway out of curiosity once the semester is over.

Assistance:

I received a lot of help on my project by using the Agent Class that Professor Allen gave us. Without the Agent, I would have had to learn to work with websites on my own. This was very helpful for me in the completion of my project.

During the project, I did not receive substantial assistance on any other portion. Professor Allen did suggest that I use a second Google Doc to store order information rather than develop an API to interface with Google Spreadsheets, but that is all the assistance that I received.