Ninja Typers Web Application

Final Project Report

By

Marvin Farrell

Contents

1. Introduction

The aim of this document is to provide an overview of the entire completed project for fourth year Software Development. It will cover area that will explain some of the major problems encountered during the project implementation and how these problems were overcome as well as detailed technical sections. The document contains a review section which highlights a personal view of what was accomplished with this project.

2. Project Description

The project that has been created is a fully automated version of a manual touch typing system that is in place at the Ninja Typers Academy. Ninja Typers (Lennon, 2012) is a touch typing club located in Carlow, Dublin and Kildare. The aim of this project is to provide a child with a interactive touch typing environment where they can improve their typing skills by advancing through a course of structured lessons that increase in difficulty by incorporating more keys into the lesson and increase the length of the lesson as the child advances. The application has a lot of added features that will be talked about in more detail in this document. The application is a web based tool that has been implemented using the following technologies Python, Flask, JavaScript, MySQL, jQuery, AJAX, CSS and HTML5. The application is running on pythonanywhere and the URL for the site tool is www.ninjatypersapplication.com.

3. Project Website

More information about this project can be found on the project website. This website contains a brief description on the project and also all project documentation produced at different phases of the project.

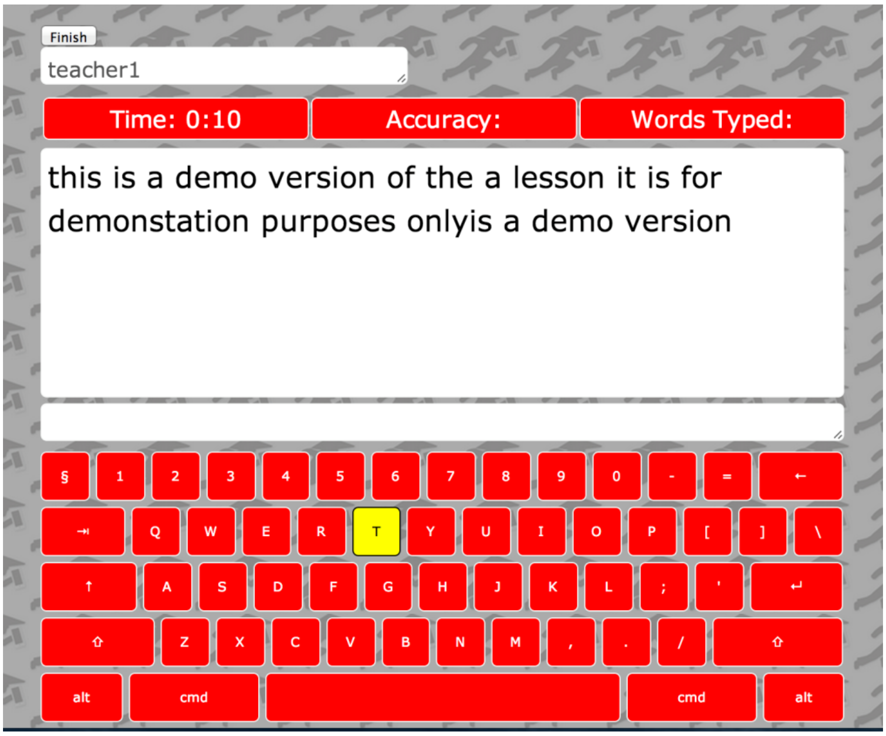
Website URL: http://glasnost.itcarlow.ie/~softeng4/C00141725/index.html

4. Main Functionalities of Completed Project

This part of the document will discuss some of the main functionalities of the project. Not all of the functionalities of the project will be included in the document.

a. Typing Lessons

The main functionality of the tool is the typing lessons. This is the page that the child using the tool performs the typing activity. All of the functionality is done on the client side. The content of the lesson depends on what grade the child is on. The lesson is retrieved from the database on the cloud and is displayed on the page for the child to type. As the child advances through the grading structure the difficulty of the lesson progressively gets more difficult by incorporating more keys and increasing the amount of the words to be typed.



teacher1

Time: 0:10 | Accuracy: | Words Typed:

this is a demo version of the a lesson it is for demonstation purposes onlyis a demo version

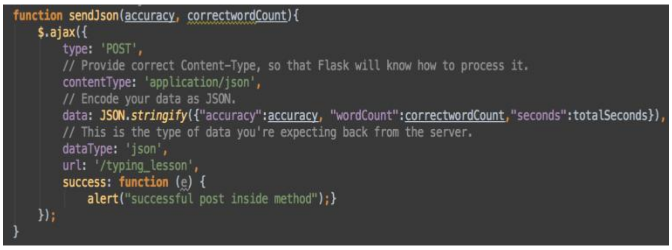
Figure 1 Typing Activity Screenshot

The time for any lesson is calculated using a JavaScript function that runs while the page is open. The time for the lesson is displayed to the user above the lesson content and updates every second.

The accuracy of the lesson is calculated using a JavaScript function. Each time a button is pressed while the function is running it fires and event which in turn invokes a function. The function used the ASCII code value for each button pressed and compares it to the ASCII value of the current position in the lesson.

The amount of correct words typed is calculated using a JavaScript function that separates each word in the lesson and compares user input to all of the words in the lesson.

When the lesson is complete an AJAX call is made to the server and this sends the data from the client side to the server side. This is code running on the back end then processes this data and updates the user account.

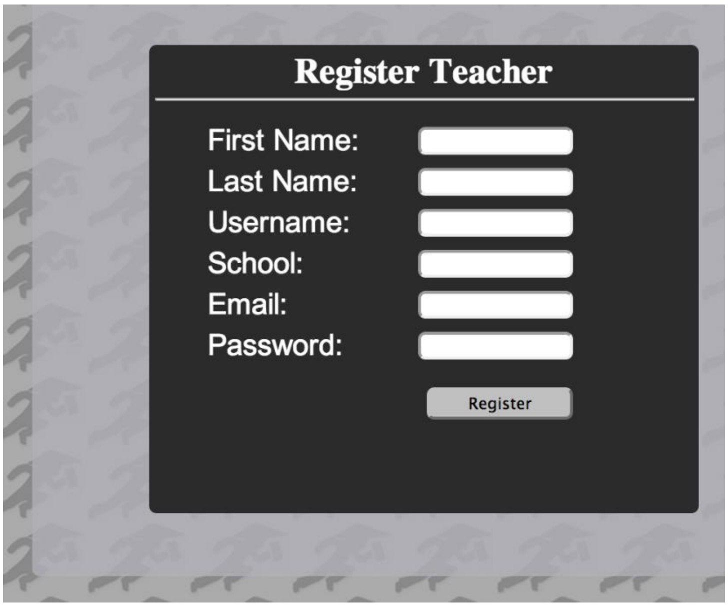


```
function sendJson(accuracy, correctwordCount){
    $.ajax({
        type: 'POST',
        // Provide correct Content-Type, so that Flask will know how to process it.
        contentType: 'application/json',
        // Encode your data as JSON.
        data: JSON.stringify({"accuracy":accuracy, "wordCount":correctwordCount,"seconds":totalSeconds}),
        // This is the type of data you're expecting back from the server.
        dataType: 'json',
        url: '/typing_lesson',
        success: function (e) {
            alert("successful post inside method");}
    });
}
```

Figure 2 Code snipped of AJAX call to server

b. Create Accounts

In order to use this application an account is needed. There are 3 types of accounts that people can register to use the application. A teacher account, a single user account and a multi user account. Both a teacher account and a single user account can be created from the web page but the multi user account must be created by a teacher and the account will be enrolled on the teacher course then. When the account is created and registered to the database it must await activating by an administrator before using the application.



Register Teacher

First Name:

Last Name:

Username:

School:

Email:

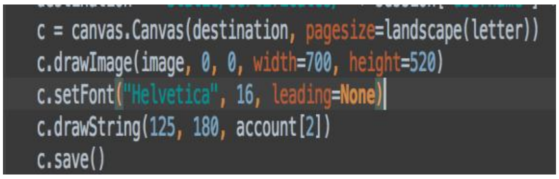
Password:

Register

Figure 3 Screen shot of teacher registration area

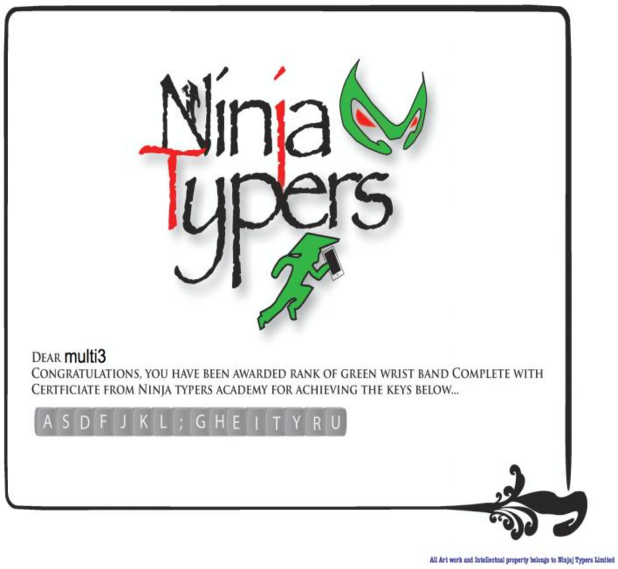
c. Download Certificates

This functionality is available to single users or teachers. A single user can download the certificate for the previously completed lesson and the teacher can download the certificates for the previous lessons completed by students enrolled on their course. The function uses the current user’s lesson grade that has been retrieved from the database to create the background image for the document. The function draws the image onto the document the overlays the users name onto the document.



```
c = canvas.Canvas(destination, pagesize=landscape(letter))
c.drawImage(image, 0, 0, width=700, height=520)
c.setFont("Helvetica", 16, leading=None)
c.drawString(125, 180, account[2])
c.save()
```

Figure 4 Screenshot of code to create custom PDF


Ninja
Typers

Dear multi3
Congratulations, you have been awarded rank of green wrist band Complete with Certficiate from Ninja typers academy for achieving the keys below...

A S D F J K L ; G H E I T Y R U

All Art work and Intellectual property belongs to Ninjaj Typers Limited

Figure 5 Sample custom PDF certificate for multi3 on green grade

Once the document has been created it is sent to the webpage which contains the download tag made available in HTML5. If it a teacher using the page a list of the students certificates are sent for download. All of the work in this functionality relies on reportlab which is a Python package for PDF creation.

d. Activate Accounts

This functionality is only available to an administrator of the site and the sole purpose of this is to activate the accounts of the people who have registered an account to use the system. The function retrieves a list of people from the database who have un-activated accounts. The administrator selects the accounts which they want to activate and activates them. As each account is being activated and email is sent to the account holder to inform them that the account has been activated. This functionality uses Pythons smtplib package to communicate with a machine across the internet.

e. Make Payments

Ninja Typers is a paid service. Single users are responsible for making their own payments while teachers are responsible for payments relating to the children enrolled on their course. Payments are made by selecting the payment option on the page and then being redirected to the PayPal (PayPal) checkout page. Upon a successful payment the user is then redirected back to the application where their accounts are updated. This process is automatic for a single user but the teacher must select which students the payment was for in order to update the account. For the purpose of the project payments are 10c rather than 10 euro.

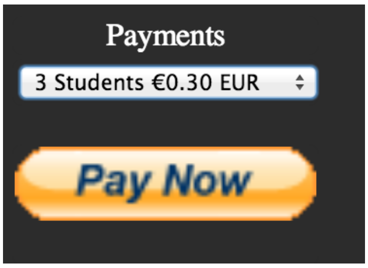


Figure 6 Payment option from teacher page

f. Create Lessons

As a teacher using the application you have the ability to create a custom lesson to be used by the children enrolled in the course. When a new lesson is created it is added to the database but it will not be used until the lesson is set to the primary lesson which distinguishes which lesson has to be picked when there is more than one lesson with the same grade.

5. Python Installation

Python comes pre-installed on Mac OSX and most distributions of Linux operating systems. To use Python on a machine running a Windows operating it must be installed manually. This was something that was temporarily needed whilst implementing this project. This section of the document highlight the steps needed to install Python on a computer running a Windows operating system.

- To download Python for Windows visit the download page https://www.python.org/downloads/windows/ on Pythons official web site.
- Download the version of Python you want.
- To install it just double click the file.
- Add Python installation Scripts folder to PATH environment variable. The location of the environment variable is Computer->Properties->Advanced System Settings ("Advanced" in XP)->Environment Variables->System Variables->Path.
- Add to path i.e. C:\Python3\;C\Python3\Scripts
- To install setuptools for Windows run the script ez_setup.py located here https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py
- To install pip run the script get-pip.py located here https://raw.github.com/pypa/pip/master/contrib/get-pip.py

6. Python Packages

There were a number of different Python packages and modules used for different areas of functionality across this project. Below is a list and a description of the packages used in the project.

a. Flask

The Flask (Ronacher, Flask, 2014) package contains the framework used from development of web applications using Python. The framework depends on two external libraries Jinja2 (Ronacher, Jinja2, 2008) and Werkzug (Ronacher, Werkzug, The Python WSGI Utiliy Library, 2014). The Jinja2 library is used to render templates while the Werkzug library is a toolkit that is used for WSGI. WSGI is an interface between web server and web applications using Python. To install flask via the command line using Linux or OSX use *pip install Flask.* The pip command can also be used on Windows but easy install and pip need to be installed first. The installation of both of these modules can be found in this section of the document.

b. ReportLab

The reportlab (Community, ReportLab, 1990-2014) module is a toolkit used for the creation of PDF documents using Python. The toolkit contains a lot of functionalities but the ones used in this project are pdfgen and pagesizes which are used primarily for the generation of the PDF and for the layout of the document.

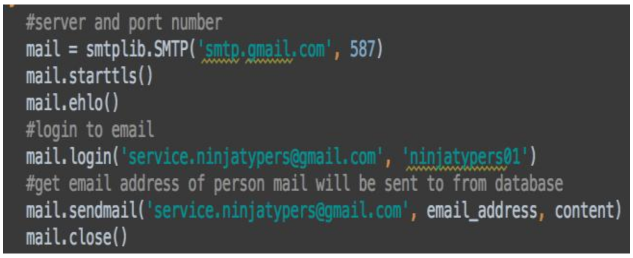
To install reportlab via the command line using Linux or OSX *easy_install reportlab* or *pip install reportlab* will work. Report lab is only supported by Python2.5 and Python2.7.

c. Functools

This python module is used to access the wraps functionality which provides the developer with pre-existing code used to create a decorator function. Functools comes as part of the Python package.

d. Smtplib

SMTP is the acronym for Simple Mail Transfer Protocol. The smtp (Community, SMTP Protocol Client, 1990-2014) module creates a SMTP client session which is used for communication between the program and any machine on the internet with an ESMTP or SMTP listener daemon. An important feature of SMTP is its capability to relay mail across transport service environments. This module is used in a two area in the project. It is used to inform an account holder that their account has been activated to use the tool. It is also used to send an email containing the password to an account holder who has forgotten their password to login. To install the package via the command line using Linux or OSX *pip install smtplib.*



```
#server and port number
mail = smtplib.SMTP('smtp.gmail.com', 587)
mail.starttls()
mail.ehlo()
#login to email
mail.login('service.ninjatypers@gmail.com', 'ninjatypers01')
#get email address of person mail will be sent to from database
mail.sendmail('service.ninjatypers@gmail.com', email_address, content)
mail.close()
```

Figure 7 Screen shot of code to send email to account holder

e. Time

This Python module contains a lot of functions relating to the date and time. In the case of this project it is used to get the current date when somebody creates an account to use the application.

f. Random

The Random python module generates random numbers for a developer to use in their own program. The most basic way to use this is to provide a range of values to pick a random number from. The random module is used in this project to generate a random lesson once the user has completed the course.

g. MySQL

The MySQL (MySQL-python 1.2.3, 2014) package is used to connect the application to the MySQL database that is used for the project. This is one of the most important packages used in the project.

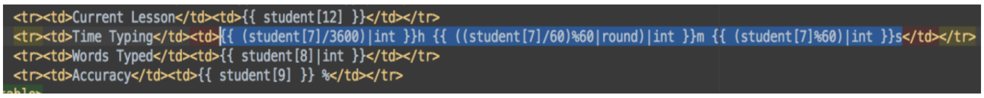
h. Installation of easy_install and pip using Windows

- Download distribute_setup.py
- Run distribute_setup.py
- Add Python installation Scripts folder to PATH environment variable. The location of the environment variable is Computer->Properties->Advanced System Settings ("Advanced" in XP)->Environment Variables->System Variables->Path.
- Add path of Python Scripts folder i.e. C\Python3\Scripts (This will include easy_install)
- To install pip use the command easy_install pip.

7. Jinja2 and Werkzug

a. Jinja2

One of the main features that flask uses in Jinja2. Jinja2 is a templating language designed for Python and is an essential part of any flask application. The main advantage of this is to be ability to use data on the server side and display it on the client side of the application.



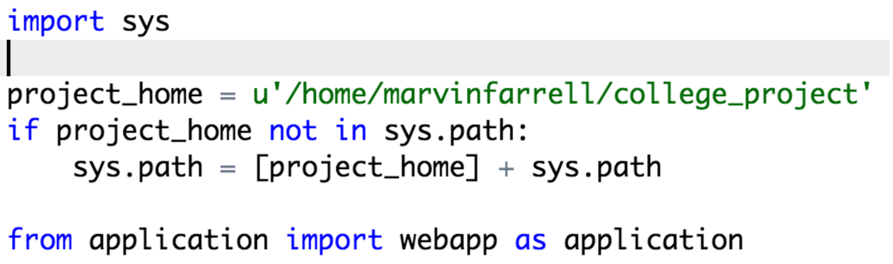
```
<tr><td>Current Lesson</td><td>{{ student[12] }}</td></tr>
<tr><td>Time Typing</td><td>{{ (student[7]/3600)|int }}h {{ ((student[7]/60)%60|round)|int }}m {{ (student[7]%60)|int }}s</td></tr>
<tr><td>Words Typed</td><td>{{ student[8]|int }}</td></tr>
<tr><td>Accuracy</td><td>{{ student[9] }} %</td></tr>
```

Figure 8 Screen shot of Jinja2 code

Mathematical operations can be performed on object values like in the example above which uses division display a numerical value in the correct time format. This was used in every single web page of the application.

b. Werkzug

One of the essential parts of the Flask framework is the Werkzug package which is a WSGI library. It is the main component responsible for routing in the application and is needed for request and response objects. Web Server Gateway Interface provides and essential interface which enables communication between a web server and a Python application. When deploying an application to pythonanywhere it is critical that the applications WSGI Python file be corrected before the application will work correctly.



```
import sys

project_home = u'/home/marvinfarrell/college_project'
if project_home not in sys.path:
    sys.path = [project_home] + sys.path

from application import webapp as application
```

Figure 9 WSGI file for project on pythonanywhere

8. Problems Encountered

This section of the document details some of the bigger problems that were encountered during the project. Each part will contain the problem, why it was happening and if a solution was found what it was.

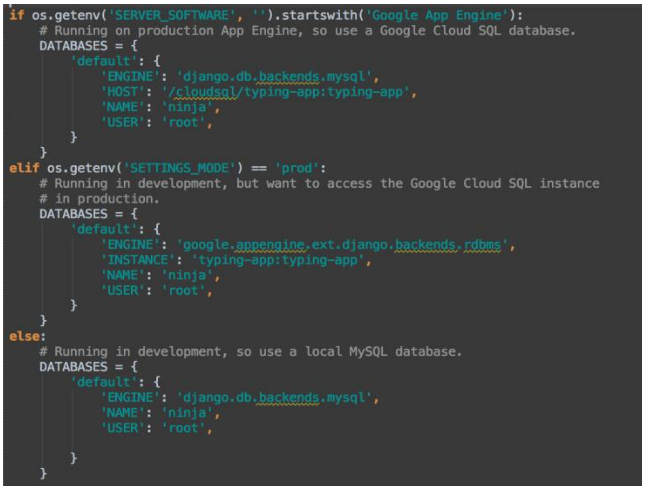
a. Django non-rel error

One of the first installs of Django was of Django non-rel. Django non-rel is a version of Django that has been modified to work with non-relational databases (NoSQL) such as the data store on Google App Engine or MongoDB, both of which could have been a possible choice for the project database. When trying to install this version of Django one of the first errors took longer than it should have to pinpoint the source of the error. Luckily after researching the error the solution was found as this was nearly an identical problem that had been encountered before. The version was running Google app engine with dev_appsever.py when it should have been running App Engine with the manage.py file with has been modified to work with Django non-rel.

b. Connecting Django to MySQL (socket problems)

Google App Engine supports MySQL databases. In order to connect a working Django application to the database an instance of a MySQL database had to be created on Google App Engine.

After an initial component of the application was developed which handled a basic account creation on the local database modifications were made to the code to connect and store the data on the MySQL instance on Google App Engine.



```
if os.getenv('SERVER_SOFTWARE', '').startswith('Google App Engine'):
    # Running on production App Engine, so use a Google Cloud SQL database.
    DATABASES = {
        'default': {
            'ENGINE': 'django.db.backends.mysql',
            'HOST': '/cloudsql/typing-app:typing-app',
            'NAME': 'ninja',
            'USER': 'root',
        }
    }
elif os.getenv('SETTINGS_MODE') == 'prod':
    # Running in development, but want to access the Google Cloud SQL instance
    # in production.
    DATABASES = {
        'default': {
            'ENGINE': 'google.appengine.ext.django.backends.rdbms',
            'INSTANCE': 'typing-app:typing-app',
            'NAME': 'ninja',
            'USER': 'root',
        }
    }
else:
    # Running in development, so use a local MySQL database.
    DATABASES = {
        'default': {
            'ENGINE': 'django.db.backends.mysql',
            'NAME': 'ninja',
            'USER': 'root',

        }
    }
```

Code changes made to connect to cloud instance of MySQL database in production and local MySQL database in development.

Figure 10 Different databases need for Django application

When these code changes after these code change were made the application had to be synced with the database using the command `python manage.py syncdb` an error was encountered which prevented the Django application from connect to the local development database.

This was the error that was continuously being produced.

```
ERROR 2002 (HY000): Can't connect to local MySQL server through
socket '/tmp/mysql.sock' (61)
```

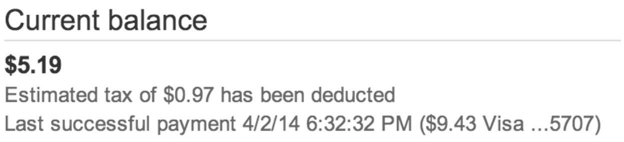
Figure 11 MySQL socket error

The first attempt to remedy this problem resulted in a change in the privileges od the users who were getting access to the database but even this had no affect or did not fix the problem.

Trying to pinpoint exactly where the problem was occurring was proving difficult until advice was sought from a lecturer regarding the issue. After about a half an hour the source of the problem was found. The installation of MySQL was not stored in the correct path so this was causing the application to look for the filesystem socket path in the wrong location.

c. Google App Engine charging for MySQL instance

Although not a technical issue this is a problem that was encountered during the development of the project that did contribute to one of the reasons from moving away from Google App Engine as a cloud platform to host the application. Google App Engine does not offer any free plans for use of a MySQL instance. An instance of a database was created for this project with the lowest costing pricing scheme which was the bottom tier and cost $0.36 a day. As these fees were charged to the bank account associated with the developers account on Google App Engine they were failing to charge the bank account which was resulting in a charge of $11.70 to the bank account for each missed payment. The minimum payment acceptable on Google App Engine is $10.00 so for anything less is a problem. The price of using a MySQL instance seemed to cost a lot more than the price outlined on the site.



Current balance

$5.19

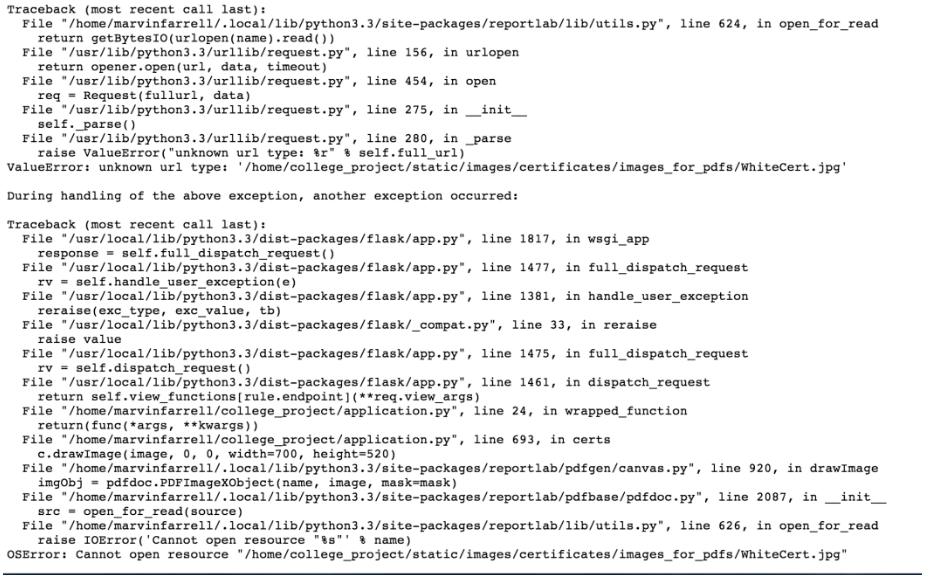
Estimated tax of $0.97 has been deducted

Last successful payment 4/2/14 6:32:32 PM ($9.43 Visa ...5707)

Figure 12 Screen shot of overcharging on MySQL instance on GAE

d. Python3.3 does not support Reportlab3.0

One of the added features to this project was to do with custom certificates. This meant that a single user could print their current certificate or a teacher print the certificates for each student in the enrolled on their course. The program created a custom PDF certificate based on the last completed grade. It used an image to create the background of the certificate and then overlaid the users name onto the certificate. It wasn't until after deploying the application did the problem surface. The functionality works when running the application locally.



```
Traceback (most recent call last):
  File "/home/marvinfarrell/.local/lib/python3.3/site-packages/reportlab/lib/utils.py", line 624, in open_for_read
    return getBytesIO(urlopen(name).read())
  File "/usr/lib/python3.3/urllib/request.py", line 156, in urlopen
    return opener.open(url, data, timeout)
  File "/usr/lib/python3.3/urllib/request.py", line 454, in open
    req = Request(fullurl, data)
  File "/usr/lib/python3.3/urllib/request.py", line 275, in __init__
    self._parse()
  File "/usr/lib/python3.3/urllib/request.py", line 280, in _parse
    raise ValueError("unknown url type: %r" % self.full_url)
ValueError: unknown url type: '/home/college_project/static/images/certificates/images_for_pdfs/WhiteCert.jpg'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/local/lib/python3.3/dist-packages/flask/app.py", line 1817, in wsgi_app
    response = self.full_dispatch_request()
  File "/usr/local/lib/python3.3/dist-packages/flask/app.py", line 1477, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/usr/local/lib/python3.3/dist-packages/flask/app.py", line 1381, in handle_user_exception
    reraise(exc_type, exc_value, tb)
  File "/usr/local/lib/python3.3/dist-packages/flask/_compat.py", line 33, in reraise
    raise value
  File "/usr/local/lib/python3.3/dist-packages/flask/app.py", line 1475, in full_dispatch_request
    rv = self.dispatch_request()
  File "/usr/local/lib/python3.3/dist-packages/flask/app.py", line 1461, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/home/marvinfarrell/college_project/application.py", line 24, in wrapped_function
    return(func(*args, **kwargs))
  File "/home/marvinfarrell/college_project/application.py", line 693, in certs
    c.drawImage(image, 0, 0, width=700, height=520)
  File "/home/marvinfarrell/.local/lib/python3.3/site-packages/reportlab/pdfgen/canvas.py", line 920, in drawImage
    imgObj = pdfdoc.PDFImageXObject(name, image, mask=mask)
  File "/home/marvinfarrell/.local/lib/python3.3/site-packages/reportlab/pdfbase/pdfdoc.py", line 2087, in __init__
    src = open_for_read(source)
  File "/home/marvinfarrell/.local/lib/python3.3/site-packages/reportlab/lib/utils.py", line 626, in open_for_read
    raise IOError('Cannot open resource "%s"' % name)
OSError: Cannot open resource "/home/college_project/static/images/certificates/images_for_pdfs/WhiteCert.jpg"
```

Figure 13 pythonanywhere error trace back

One of the advantages of using pythonanywhere is the different logs associated with each project. In the error log it was slightly tricky to fully debug the problem but it did the source of the problem. Further research into the problem eventually led to Reportlabs PyPi page. This states that Reportlab only supports Python2.5 and Python2.7. Reverting back to Python2.7 was causing error in more than one area of the project so it was decided to stick with Python3.3.

e. JavaScript Debugging

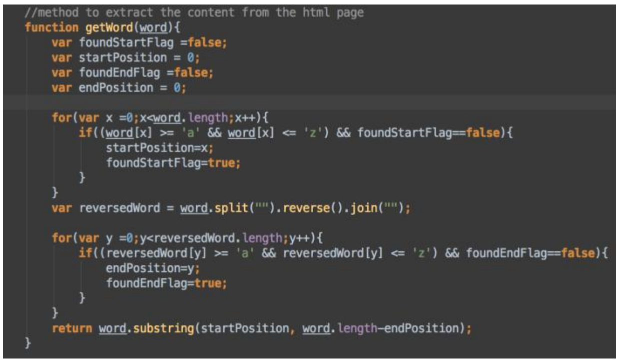
This was a tricky area of the project. Having no experience with JavaScript it was very difficult to pinpoint the source of an error that was preventing a while script from running on a page. The methods used at first were inserting console logs and alert at different points of the code. The console.log (e) method is used to log “e” to the browsers console, using an alert (e) will trigger and alert message with “e” being the message. This was a quick hack to find out where the problem area was. It wasn’t until towards the end of the implementation of the project that Firebug was found. Firebug is a web development tool which can be used for debugging.

f. Updating Accounts after lesson

Creating an algorithm that carried would carry out the correct steps to update a user’s account after a typing lesson required a lot of time and planning. There is a lot of different cases in which the users can be updated. This took longer than expected to create because there are many different contributing factors that make this algorithm slightly larger and more complex than originally thought.

g. Passing data from Python to JavaScript

Passing the content of the lesson from Python to a JavaScript function using JSON objects proved a lot more difficult than anticipated. Ideally a JSON object would be posted to the server and an AJAX call would also be made. This was difficult to do without posting just that data to a page. By combining JavaScript and Jinja2 templates solution was found. When the original data from the template was retrieved it was formatted differently and extra characters seemed to be added to the value. The way around this was to take the value from the template and use a JavaScript function that was implemented to extract only the data that contained the lesson content.



```
//method to extract the content from the html page
function getWord(word){
    var foundStartFlag =false;
    var startPosition = 0;
    var foundEndFlag =false;
    var endPosition = 0;

    for(var x =0;x<word.length;x++){
        if((word[x] >= 'a' && word[x] <= 'z') && foundStartFlag==false){
            startPosition=x;
            foundStartFlag=true;
        }
    }
    var reversedWord = word.split("").reverse().join("");

    for(var y =0;y<reversedWord.length;y++){
        if((reversedWord[y] >= 'a' && reversedWord[y] <= 'z') && foundEndFlag==false){
            endPosition=y;
            foundEndFlag=true;
        }
    }
    return word.substring(startPosition, word.length-endPosition);
}
```

Figure 14 Screen shot of function to extract lesson content from web page

h. Creating a CNAME

Pythonanywhere (pythonanywhere, 2014) allows you to run one application on the site before you are required to sign up for a developers account. When you are running a number of applications you have to have a custom domain name so that your URL can point to the application hosted on pythonanywhere. A CNAME (Canonical Name Record) is needed for this. In order to create a CNANE a domain had to be purchased and the record created on the DNS provider. The domain for this tool was purchased off

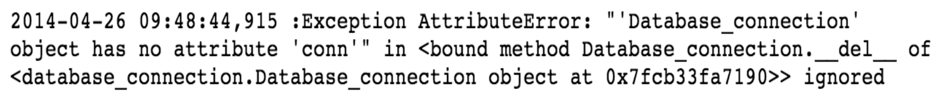
BlackNight Solutions (Blacknight Solutions). There were some issues trying to set up the CNAME on the site but it was eventually solved by removing one of the A records which points a hostname to a server by the servers IP address.

i. Keyboard Bug

One bug that was discovered after deployment was a JavaScript issue. A function was created to change the colour of the next key to be typed from red to yellow. There is a bug that prevents this from happening if the next key is the same as the current key i.e. if a word see is to be typed the first “e” key will change colour to yellow but then when it changes back to red the next “e” will stay red also and not change colour.

j. MySQL database on pythonanywhere

Deploying the application to pythonanywhere was a smooth process but problems were encountered when trying to connect to the database that was created for the project that was also on pythonanywhere. It was failing to connect and this was preventing the application from reading or writing to the database.



```
2014-04-26 09:48:44,915 :Exception AttributeError: "'Database_connection'
object has no attribute 'conn'" in <bound method Database_connection.__del__ of
<database_connection.Database_connection object at 0x7fcb33fa7190>> ignored
```

Figure 15 Database connection error on pythonanywhere

Debugging and researching the error leaded to this error being fixed. There were a number of small problems that was causing this error. One of them was that when a database is created on pythonanywhere any reference to that database must concatenate the pythonanywhere account name with the database name. In this case the account name is “marvinfarrell”, the database name is “ninjadb” and the resulting reference to the database is “marvinfarrell$ninjadb”.

k. Jinja2 Inheritance

One area that produced unexpected problem in this project was inheriting from parent templates using Jinja2. It was an area that I had previously worked on but it seemed to cause problems during the Construction Phase of the project. Not able to find the exact source of the problem the templates were created independently. This has no effect on performance or functionality it just meant producing more code per template.

9. Algorithms

This section of the document contains some of the algorithms used for some of the more complex areas of the project.

a. Login

- Get username and password from page
- Get account from database using username and password
- If account is not empty then

    - If account is activated then
        - Set session[“username”]
        - Set session[“password”]
        - Set session[“usertype”]
        - Set session[“logged-in”]
        - Redirect to user homepage
    - Else
        - Redirect to pending account page
- Else
    - Redirect to account not created page

b. Change Lesson

- Get lesson name and lesson grade from page
- Get all lessons from database
- Check all lessons for lesson name retrieved from page
    - If lesson name is found then
        - Set found flag to 1
- If found flag is 1 then
    - Redirect to Change Lesson page
    - Update all accounts with same lesson grades primary_lesson field to 0
    - Update lesson with same lesson name primary lesson_field to 1
- Else
    - Redirect to Change Lesson Page

c. Update account after lesson

- Get current users account from database using username and usertype cookies.
- Get current lesson from database with current users current lesson
- If current user is not finished the course then
    - Set lesson to current lesson
- Else
    - Pick random lesson
- Request data from server
- If data is not empty then
    - Get accuracy from server
    - Get words typed from server
    - Get time from server
    - Assign accuracy to session[“typing-score”]
    - Assign words typed to session[“words”]
    - Assign time to session[“time”]
    - Calculate new scores with scores from lesson with scores from account on database
    - If accuracy from lesson is greater than lesson score then
        - Update user account scores
        - Update to new lesson
        - Redirect to lesson summary page

```
o Else
    ▪ Update user account scores
    ▪ Redirect to lesson summary page
```

d. Calculate Average and Words Typed

```
• While x < lesson.length
    o add button pressed to word variable
    o If button pressed value == lesson.charCodeAt(x) then
        ▪ Correctchar ++
    o If button presses value == space
        ▪ Get substring word variable to x
        ▪ Compare with words from the lesson
            • If words match then
                o Correctwords ++
    o Charactercount ++
    o X ++
    o accuracy=((1/characterCount)*correctChar)*100;
```

10. Meetings

I had to attend a number of meeting with Keith Lennon the owner of Ninja Typers over the scope of this project. These were important meeting for me to receive feedback from Keith and for Keith to see what work was being carried out on the project. This section of the document is a record of each of the meetings I had with Keith and what the subject matter of the meetings were.

Meeting 1

Date: 10/10/2013

I met with Keith Lennon to establish where we were going with this project. The first thing we discussed was his vision of Ninja Typers and where he was looking to take it. We talked for about 20 minutes about this before even talking about the project. When we started to talk about the project I took notes on the different functionalities that he wanted incorporated into the application. I then outlined what I would be competent enough to implement and also outlined the areas that might pose as potential problems areas for me. We set up a shared DropBox folder which would be a central location for us to share all documentation and resources for the project.

Meeting 2

Date: 19/10/2013

I visited Ninja Typers on a Saturday morning when there was a class going. There were a number of reasons I wanted to visit the class and below are some of the observations that I took from the class.

- Manual System

  The current system that was in place is very simple yet very effective. It is a word processing application with the lesson for the student typed at the top of the page. The student has to retype all of the content on the top of the page a number of times to complete the lesson. There is no error checking in place for this type of system. At the end then it is easy to compare between the text at the top of the page and at the bottom of the page to see what (if any) mistakes were made when typing.

- Workload on Keith
  There were 10 children attending the class the morning I visited Ninja Typers. This is the maximum number of children that can attend the class at a given time. Any time any of the children had a question or a problem Keith would have to tend to them. A number of times whilst helping a child he was required by another child. This seemed like a lot of work to be doing over the course of the hour.
- Attention of Children
  When chatting to Keith he had informed me that a number of the children that were attending Ninja Typers had different learning disabilities and mental health problems. Some suffered with Autism, Dyslexia, Asperger Syndrome, ADHD and Dyspraxia. Children with these attributes have harder times concentrating and spending all their time doing the lesson without stopping or getting easily distracted.

- **Meeting 3**
  Date: 18/11/2013
  We arranged to schedule a meeting to discuss different technologies that could be used in the project. I showed Keith a list of the use cases that I had come up with for the project. We modified these use cases and came up with some new ones to suit the project. We also done some rough sketches of the possible main typing activity display to get more of a direction on where to go with it and to try come up with an initial prototype design.

- **Meeting 4**
  Date: 09/01/2014
  I arranged a meeting to discuss problem areas in the project. I also wanted to show him some of the working components in the project. We made some modifications to the display of the main typing activity to remove some things on it which may distract a child taking a lesson.

- **Meeting 5**
  Date: 01/04/2014
  Keith wanted to meet up to see where we were at with regards to project completion and schedule.
  We discussed different areas of the project that we might change and some features to add it. We did some basic test on how a child might use the application. I showed him the backend of the application and how it was interacting with the database to retrieve the correct information for the current user. He seemed extremely happy with the project which was a huge boost for me to know that I was going in the right direction with it.

11. Conformance to Specification and Design

An attempt was made to follow an Agile Development methodology over the scope of this project. The reason for this was to allow a change in requirements and design at a later stage in project. This part of the document details some of the major changes from that were made from the ideas put forward in the Functional Specification and the Design Document.

a. Framework: Django to Flask

Originally Django was chosen as the framework for the project as Django seemed like it was capable of handling all aspects of web development. It was chosen over Flasks which initially it was thought that it was not a large enough framework to handle a project like this. Eventually Django was dropped and Flasks was chosen as the Framework for the project.

Reasons:

- **Lack of web framework knowledge**

  During research of the project and especially researching web frameworks it was hard to weigh the pros and cons of one web framework over another without a real understanding of them. At first with was initially thought that Flask being a micro-framework was too small and not be capable of supporting a project like this. The main reason is that this was the first bit of web development I have ever undertaking. Django was chosen because it ticked all the boxes for this project.

- **Development Difficulties**

  During development using Django a lot of problems started showing up trying to extend the User Registration class and make custom forms for user accounts. This was the first difficult problem to overcome using the framework. Another problem area arose with database issues and trying to connect to a non-local database. With it being such a large framework there is just too much to learn especially with it being the first web framework for a developer to use.

- **Deployment to cloud**

  This was a huge contributing factor to the changing frameworks. Deploying Django to Google App Engine was extremely difficult because it also had to connect to a MySQL instance also. Each time and error was fixed it seemed like the next error was more difficult to debug and fix. This was taking up way too much time. A successful deployment of a flask application made spending so much time trying to deploy a Django application like pointless.

- **Google App Engine Support**

  After implementing a couple of components using Django it was discovered that Google App Engine support up to Django 1.5. The version that was being used was Django 1.6 which had a lot of features that were being used in this project that Django 1.5 did not support.

b. Cloud Storage: Google App Engine to pythonanywhere

pythonanywhere was a cloud platform that was not found when carrying out research for this project. It wasn't until halfway through the project that it may have been a better suited framework for the project. It was decided after some discussion to drop Google App Engine and pick pythonanywhere as the cloud platform for the project.

Reasons:

- **User Interface**
  pythonanywhere offers a much better user experience that Google App Engine. It has a console page which provides the user with two types of console. A bash console and a MySQL console. Some big advantages to this are installing python modules and using a MySQL schema upload by the developer to create a database on the cloud.

- **Support**
  Forums on pythonanywhere are an excellent place to get answers to any issues regarding pythonanywhere. A lot of people are genuinely interested in helping people, which is very beneficial to somebody who is new to the site.

- **Cost**
  There is no cost associated with using pythonanywhere. If a developer wished to have multiple applications running on the site it does require a developer account which cost $12 dollars a month. For the service that is provided this fee is very good considering that on Google App Engine you are required to pay for every MySQL instance that is running on your account.

- **Deployment**
  The process of deploying an application to pythonanywhere seems to be the most user friendly way possible. Below is a list of steps initially undertaken to deploy the application to pythonanywhere.

  1. Compress project folder
  2. Upload project folder to pythonanywhere
  3. Open bash console and unzip compressed project folder

4. Change all references to local database to database on pythonanywhere note: any refrence to a database on pythonanywhere requires a combination of the account name and database. In the case of this project it was “marvinfarrell” and “ninjadb” and the resulting reference was “marvinfarrell$ninjadb”.
5. Modify WSGI file on pythonanywhere by changing app to the application name

12. Project Review

This section of the document is my own personal reflection on the project. It covers what I took away from the project, what I was not happy with and also what recommendations I would make to somebody who wishes to implement a similar project.

a. Unsatisfactory Areas

The styling on the applications pages could have been a lot more crisp and professional looking if more time have had been spend on the appearance of the project. Although the functionality behind each of the pages is working as expected the appearance is what the user can see and this is something that could have been improved.

b. Uncompleted Areas

○ JavaScript Pointer

A functionality that was not complete was a small feature with the typing activity page. The content for the lesson displayed over the keyboard is easy to read, however when the user is typing it is easy to get lost as to what the current character is to be typed. Creating a JavaScript class to do this seemed to pose a problem which ultimately was not overcome. Ideally the function would have highlighted the current character with a border.

An alternative to the way the lesson was displayed on the page could have been in table format which would have supported a script that would alter the current table cell to provide the user with a visual aid when taking the lesson.

○ Data Encryption

Another area that was not complete was data encryption. It was assumed that pythonanywhere would automatically encrypt data stored on a MySQL database on the site. This was not the place and the only security measure in place on the database is password protection. This is an area that should have been uncovered during the research stage of the project but unfortunately wasn’t leaving any time to implement other security measure towards the end of the project.

c. Changes I would make if I had to start again

If I was to start this project again I with a better understanding of what the project entails I would have defiantly started the implementation at a much earlier stage. The way I approached the construction phase was completely wrong. I used a big time window to progress at a moderate pace which I think was a bad idea. If I had spent more time earlier on coding it would have also been a chance to validate the technology choices I had made and would have been able to deal with some of the framework problems I had at a much earlier stage.

I would also have carried out a different approach to the way I carried out the research to the project. The way I did approach it was to source information from books and the internet. The way I should have done the research an area, do some small tests on that area. A good example of this would have been in regards to deployment of an application to the cloud. Had a tried to deploy even a basic application like a “hello world” app to Google App Engine and to pythonanywhere it would have given me a better indication as to which one to pick and would have saving me time at a later stage of the project.

d. What was learned?

○ Technical

○ Web Technologies

This was the first web based project I have completed. There was a lack of knowledge around web technologies at the start of this project that has gradually increased the more work I done on the project. One of the biggest things that I learned with regards to web technologies is linking the server side of an application to the client side of the application. The idea of passing data between two separate components was slightly intimidating because it was an area that I knew little about. By properly researching the area it led me to using AJAX calls and JSON to exchange data between server and client side.

○ Languages

Some of the technologies used in this project are Python, Flask, MySQL, JavaScript, jQuery, AJAX, HTML and CSS. This project was the first time I have been exposed to the majority of the technologies listed. Although difficult at times to code in different languages and to debug I learned an extraordinary amount in the timeline for this project. These are attributes that I have added to a technical skillset that can only seem to grow and expand. By using some of these languages it has also exposed what some of the technologies are capable of, why they might prevail over others in a problem domain and why they are so widely used.

○ MySQL and SQLite3

I gained a lot of experience of the course of this project using MySQL and SQLite3 databases which is an essential attribute to have going into the working environment. Prior to this I had no experience with working with relational databases hosted on a cloud platform.

Cloud Computing

With cloud computing rapidly increasing it was crucial for me to experience it during my project. I was able to try out two different cloud solutions Google App Engine and pythonanywhere experience why one platform may be more advantages over the other for the type of project in question. It was good to see the difficulties in trying to deploy an application to the cloud and trying to fix the problems associated it.

Incremental and Iterative Development

Although I had researched this area I have never applied it to any project that I have undertaken. This was an area that was not even considered to be a part of the project but as implementation started it was clear that this was the method that was being followed. By using customer feedback from Keith Lennon and gradually adding features to the project I was able to incorporate this aspect of Agile Development into the project. Each of the four phases of this development method has been incorporated into this project.

- Inception Phase: Functional Specification, Research, Vision
- Elaboration Phase: Designing the system
- Construction Phase: Implementation of the system
- Transition Phase: Deploying to a pythonanywhere
-

Version Control

BitBucket (BitBucket) was the site that supports Git (Git (software)) version control which was used for this project for source code management. This is a very important part of software development that proved very important in this project. All changes made were pushed to a master copy and each push was noted. This was vital when working on area of functionality in the project that could have had an effect on another area of the project that is working as intended.

Personal

Research

This is often an area that is overlooked but I think it is an essential thing to have in the skill set of a Software Engineer. At the begging of the project I didn’t think that carrying out good research would be an important factor in this project but it was and it ended up costing me time that I could not get back. The quality of the research that I would do now would be at a much higher standard than that of the research I would have carried out at the start of the problem. There are 3 contributing factors to this.

1. Time: Set out a timeline to find information relating to the topic you are researching.

2. Resources: Take note of the more resourceful places you find information. Any problem you are having is likely to have happened in the past and it would have been documented.
3. Research Area: Try to pinpoint exactly what you are trying to research. Set out different areas to research that will lead to more efficient research.

○ Time Management

Very crucial aspect of this project was time management and sticking to the timeline set out in the project plan document. Had it not have been for the plan and the milestones set out during the planning of the timeline it would have been very easy to fall behind and not complete the project by spending too much time on one task. Some tasks did take longer than expected and other shorter than expected which provided a balance. At most the project ran behind schedule by about 2 weeks over the Christmas period but by prioritising a workloads and different tasks the project got back on schedule. This was the first project that I implemented that had a number of different milestones to reach and was able to do so with good time management.

○ Communication

A very important part of any software project relies on how satisfied the customer is with the software that is being developed for them. I was fortunate to experience this during my final year project. I had regular meeting with the person that I was developing this piece of software for. These meetings allowed me to stay on track and develop the software exactly how the person wanted as well as adding requirements and extra functionalities that may not have been mentioned in the Formal Specification at the earlier stages of the project. I think that gaining exposure to this area this early on and using customer satisfaction to advance a project was a valuable learning experience for me and really showed the importance of a good communication channel between a developer and a client.

○ Following a Software Development Model

This was the first time a development model was applied to a project I was involved in. By applying the characteristics and producing required deliverables at each phase of the unified process I was able to develop a fully functional software system.

○ Inception Phase:

- Business Case: A business case was put in place to highlight the potential for this product in a market and also who the target users of the product could be.
- Project Risks: Some of the potential risks associated with a product like this were establish.
- Use Cases: A model was used to visually represent what each user can do with the system. This model was an overview of some of the detailed use cases highlighted during this phase.

All these areas were documented used to produce a Vision document which is the first document to be produced during the Inception Phase of the Unified Process. Each of the use cases outlined in this stage has been fulfilled in the project. The biggest risk outlined proved to be one of the bigger problems encountered over the scope of the project.

- **Elaboration Phase:**
    - Research Document: The next area was to carry out research into the different technologies that could be used in the project. Over the course of the research many technologies and many areas were documented and studied to influence the decision to pick the right technologies for the project. All findings and choices were recorded into the Research Manual document.
    - Functional Specification: The requirements for the system were the next area of the project that were analysed and the use cases were refined to make them more accurate. Similar applications were looks at in terms of design and usage and also all the external interfaces that would be used by this project were outlined. All this was recorded into the Functional Specification.
    - Design Document: Designing the system architecture and individual components was the next stage of the project. Class diagrams and databases had to be designed to work with this system. The framework for the project was picked and the project had to be designed around that. An appropriate software stack and some design for some of the pages of used in the project are all documented in the Design Document.

All of these are important areas that make up the Elaboration Phase of the Unified Process. Each of these area were key areas in this project

- **Construction Phase:**

This was the longest area of the project. Each of the features outlined in earlier phases was implemented. This phase lasted about 15 weeks. Deliverables for this phase were increments each of was adding to the system.

- **Transition Phase:**

The main goal of this area was to deploy the software system to a cloud platform. Although one feature of the project did not work this is an area that was successful. The project is now live on the internet and in production waiting to be used by members of the Ninja Typers Academy.

By carrying out a number of different tasks at different stages of this project it has allowed me to use the Unified Process model and use incremental and iterative development in a software project. It was the first time I have seen how all these different area are used in software development to produce a functional software system. The timeline for the Unified Process is nearly identical to the timeline and plan that was adhered to over the scope of this project.

e. Were the technology choices the right ones?

After changing from Django to Flask for my web development framework and changing from Google App Engine to pythonanywhere for my cloud storage and hosting I would say that the technology choices that were made to implement this project were perfect. Flask is regarded as a micro-framework but it is an extremely powerful web development framework. Because most of the technologies used in this project were new to me I wasn't fully able to maximise the capabilities of the technology but was able to use it them to improve the quality of this project.

f. Recommendations to someone attempting a similar project

○ Research

One of the biggest contributing factors to the project success will be the breadth and quality of the research that is carried out relating to the project. This may not seem like one of the biggest aspects of the project but it is only after completing a project do you truly understand how important it is and how much time it will save you if you research an area properly.

Use the research to influence the choices for the technologies used in the project by testing them on a small scale at an early stage. Don't pick the most popular option because you think it may be the easiest. Sample the technologies uncovered in the research to see why they might suit the project the most.

○ Time Management & Planning

The timeline in your project plan should be visible to you every day. This is an excellent indication as to where you are at in your project, if you are on schedule and what work still needs to be done. Be sure to adhere to the timeline you set out and prioritise your workload. Develop a realistic timeline that deals with the more difficult area first and that will maximise the chance of project completion before the submission deadline.

○ Early Incremental Development

Start coding early. Use the information that was uncovered during the research to start coding at an early stage of the project. By doing this you are creating chances to validate technologies that you are using. It creates a good time window to change technologies and not have a detrimental effect on the project as a whole.

13. Acknowledgements

There are a few people I would like to thank for their help over the course of this project. First of all I would like to thank my classmates who have been there to help in whatever way they could. Next I would like to thank Joseph Kehoe for the planning of our demos and Paul Barry for assistance with a problem that I was struggling with that was preventing me from advancing forward with my project. I would also like to thank Keith Lennon for taking the time to meet me when I had an issue regarding the project. Finally I would like to thank Dr Chris Meudec who has been a tremendous supervisor to guide me over the course of this project. Chris planned regular meeting with me where he went through every area of my project. Had it not been for Chris I probably would not have completed my project. So again I would like to massive thank you to everybody that has help me, it is very much appreciated.

14. Conclusion

Overall I would class this project as a success. Everything that was outlined in the early stages of the project was achieved and a lot of extra features were also added. By following the structure put in place by Dr Chris Meudec I was able use the Unified Process in a software project. Theoretically understanding the Unified Process was something that before the project I couldn't really grasp but since completing this project I can now see the importance of each of the stages and how each is a massive contributing factor to a project as a whole. This was an excellent opportunity to get involved in meetings and use customer feedback to proceed into the project. It has been a lot of responsibility to produce a project like this and meet deadlines and produce a deliverable which is something that can be classed as a success. The biggest thing that I can take away from this project is with regards to the technologies used. A lot of new technologies and areas of software development have been exposed over the scope of this project especially client side and server side development. This has given me a much clearer understanding of the direction I want to take as I start my career as a Software Developer.

15. Bibliography

MySQL-python 1.2.3. (2014). Retrieved April 17, 2014, from Python.org: https://pypi.python.org/pypi/MySQL-python/1.2.3

pythonanywhere. (2014). Retrieved February 2014, from pythonanywhere: https://www.pythonanywhere.com/

BitBucket. (n.d.). Retrieved April 22, 2014, from Wikepedia: http://en.wikipedia.org/wiki/Bitbucket

Blacknight Solutions. (n.d.). Retrieved March 2014, from Blacknight Solutions: https://www.blacknight.com/

Community, P. (1990-2014). *ReportLab*. Retrieved March 2014, from Python.org: https://pypi.python.org/pypi/reportlab

Community, P. (1990-2014). *SMTP Protocol Client*. Retrieved March 2014, from Python.org: https://docs.python.org/2/library/smtplib.html

Git (software). (n.d.). Retrieved April 20, 2014, from Wikipedia: http://en.wikipedia.org/wiki/Git_(software)

Lennon, K. (2012). *Nnja Typers*. Retrieved January 02, 2014, from Ninja Typers: http://www.ninjatypers.com/

PayPal. (n.d.). Retrieved February 2014, from PayPal: https://www.paypal.com/ie/webapps/mpp/home

Ronacher, A. (2008). *Jinja2*. Retrieved February 2014, from Jinja2: http://jinja.pocoo.org/docs/

Ronacher, A. (2014). *Flask*. Retrieved December 2013, from Flask.org: http://flask.pocoo.org/

Ronacher, A. (2014). *Werkzug, The Python WSGI Utiliy Library*. Retrieved March 2014, from Werkzug: http://werkzeug.pocoo.org/