MapMyNotes

Final Report for CS39440 Major Project

Author: Ryan Gouldsmith (ryg1@aber.ac.uk)
Supervisor: Dr. Hannah Dee (hmd1@aber.ac.uk)

4th March 2016
Version: 1.0 (Draft)

This report was submitted as partial fulfilment of a BSc degree in Computer Science (G401)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work I understand and agree to abide by the University's regulations governing these issues.

Name ....

Date ....

Consent to share this work

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name ....

Date ....

Acknowledgements

I am grateful to...

I’d like to thank...

Abstract

Include an abstract for your project. This should be no more than 300 words.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Evaluation

This chapter will evaluate how well the project as a whole has gone. It will look at different issues relating from correctly identified requirements, design decisions, what was good about the project and areas which could be improved.

1.1 Correctly identified requirements

To recap, the following requirements and objectives were identified at the beginning of the project:

1. Investigate how to extract handwriting text from an image - this will involve looking into ways OCR tools can interpret handwriting.
2. Train the OCR to recognise text of the author's handwriting.
3. Produce a set of a rules which a note must comply to.
4. Produce a web application to form the core part of the product. This includes allowing a user to upload an image, display the image. Add appropriate tagging to a note such as module code.
5. The user must be able to search for a given module code, showing the fill list of notes based on the module code entered.
6. The backend of the application must conduct basic OCR recognition, analysing the first 3 lines of the notes.
7. The backend must integrate with a calendar to archive the notes away later to be found again.

The afformentioned requirements were part of the core system requirements; these were classufied as the minimum funtionality of the application. Personally these targets, which was partly self-imposed, was extremeley challenging. The amount of work that was taken on, with lots of different components required that work was produced at a steady pace.

Investigation and research work was extensively conducted to identify how handwriting can be extract from an image. This was further enhanced by looking to optimise the performance of the Tesseract engine. Not only was the image binarised - there were studies and iterations of improvemet which would help to benefit the end user.

After twelve different training examples, from hand, the Tesseract enging is not yielding any greater of a success. Therefore it can be concluded that a sufficient job has been achieved with handwriting training. Probably one of the most frustrating experiences with labelling individual characters. Nevertheless, the training could not return a greater success than currently therefore good work has been completed to get it to that stage.

With the web application being the core part of the problem, then a lot of effort was centered in on ensuring the correct functionality was added. Personally, the application reflects that of the work carried out on it. A user can upload an image to the application and tag all associated meta-data, including the title, module code, date, and location.

The application creates an easy to use interface that allows the user to search for a given module code with ease. The case in which they enter the text is not important, as the application handles the case-sensitivity allowing for a better user experience.

One of the more impressing features of the application is the handwriting recognition integration. It parses the first three lines of the image and shows it to the user in an intuititve manner. This was built ontop of the work already implemented in the project with the handwriting recognition. With the final steps included the application has its true purpose, and that extra bit of complexity which personally the application needed.

Finally, from the original requirements the calendar integration was succesfully implemented; it was in retrospect one of the more challenging tasks to overcome. Ranging from the testing the services to implementing reocurring events, Google calendar always threw up a lot of issues. The reocurring events and all day events were particially frustrating, however there has been a wide interaction with the calendar. It can list all events in the last 7 days, find specific events by day and is able to find reocurring events. Ontop of this is can add to an event description and remove a specific string from the description field.

Overall, the application produced meets all the core requirements and goes beyond, in some aspects, the original features. The app has been well tested with a strong testing infrastructure backing the application. There have been complex design decisions to ensure that the design of the application is simple and intuitive to a reader. Personally, the app has been completed to a good standard.

1.2 Design decisions

Reflecting on the design decisions which were evaluated in section x. The use of the MVC framework structure was extremely beneficial. It encouraged that the code was decoupled at all times, leaving the system extremely modular. Spending the time to construct a good MVC structure was worth it.

The overall database diagram is well considered, and each of the relations is correctly identified and considered. One aspect which would be nice to change would be the note relation, so that the image path was represented in its own relation.

Additionally, there are a few functions in the binarise image class which should be static.

Overall, the design of the classes show a good level of understanding regarding the object oriented paradigm. The use of helpers and service objects help to extend the system, reducing the amount of duplicated code, whilst semantically collating similarily related code together.

Although not strictly related to the core design of the system, but it was wished that the PEP8 [CITE] standard was followed and implemented as the application was being developed. Due to inexperiences using Python, this was not realised until later on in the project - this resulted in a larger refactor and reflecting it should have been conducted at the very beginning.

1.3 Use of tools

1.3.1 Flask

During the project there were times in which there were feelings that the toolset should be different. Flask, the micro-framework for the web application, was pehaps a bad decision in hindsight. There was a lack of documentation and out-the-box support made even the most basic of functionality difficult and uncessarily complex. The configuration files for example: there was no default support for multiple support for default configuration files.

Another example is default security concerns. As stressed the importance of in *Internet based applications* cross-site request forjery (CSRF) is a big issue among internet security. By default Flask does not support CSRF checking. An additional 3rd party library, SeaSurf, had to be installed to handle the basic functionality of this.

That said, Flask did offer a great deal of flexibility regarding the project. Directories were customisable easily, and the use of blueprints enabled that MVC feel to the project. There were times which just the basic support for a framework would have aided the developer and helped to improve the productivity, rather than being subjected to more external code.

1.3.2 OpenCV

The choice of using OpenCV compared to ImageMagick proved to be an extremely important design change. This helped the Tesseract training to increase substaintially, without the use of OpenCV it cpuld be argued that the Tesseract training would never have gotten off the ground. The research conducted into looking into different binarisation techniques was imperitive; there could have been a long chase going down the greyscale image route, trying to train the data and not getting anywhere.

1.3.3 PostgreSQL

The use of PostgreSQL was overall a good choice. Potentially all the decisions regarding using that over MySQL were not fully met, restrospecitively analysising the decision. However, the decision to use that over SQLlite, especially as more than one person would use it would be a wise decision that was right to be made.

However, one aspect which should be noted was that if the project was to start again then a more serious look into the NoSQL database should be considered. During the application development, it was acknowledged by Dr Hannah Dee that it would be good to use the application for conferences too. The conferences do not follow the same structure as that of a lecture, prodominetly lecture name and module code.

Due to the variability in the differing data returned for different event then prehaps a NoSQL approach would have been beneficial.

1.3.4 Tesseract

Tesseract itself has been a great tool during the creation of the product. The training process was a little tedious but the tool would not be swapped out for another form of OCR tool.

1.3.5 Google Calendar

Although Google Calendar threw up a lot of issues along the way, the choice was a good decision in the end. After analysing the calendars choosing the most popular one appealed to more of the market.

Additionally, all the good support that Google gave was beneficial in the end, due to a large community of developers available.

1.4 Meeting the users needs

One of the key premises that the application was worthwhile was that the user would need to digitise their notes easily. However, from the analysis given back from the user testing it seems as though that selection of users did not find that it would improve their note-taking abilities.

Primarily, this was due to people taking notes in different ways. They often prefer to write up their notes. One way in which this could be overcome would be to provide a WYSIWYG editor, so the user can have full control over what was achieved and entered into the note application.

So technically, the application did not meet the use-case of the users wholly. Certain aspects of the application would have to be developed further, to ensure that the users needs were fully met. However, due to personal preference with note-taking then meeting the needs of all the users would be unlikely, hence why there’s more than one solution available for the same problem.

1.5 Additional project aims

Due to the nature of the project, all the features and functionality could not be implented into one project spanning just 15 weeks. Due to this a few requirements were cut back from the initial specification.

If this project was taken further then additional features would be most welcome.

1.5.1 Auto-correcting of image

A big problem with the handwriting recognition is that the image has to be correctly rotated to 90 °, and cropped sufficiently. This would have been a nice feature to look to implement, but due to the time contraints this was not plausable. Sadly, the resultant of not including this will impact the user-experience, as it would have to be cropped by hand.

This is something what time was spent implementing, but issues with the calendar arose.

1.5.2 Extracting images

A nice feature which would have been useful would be to extract any images from the notes. This would mean that the notes woul have to have a massive restructure with the way it parses text. However, been able to select the notes and then change the size on a canvas would have been a great feature to implement.

1.5.3 Generic handwriting recognition

A core acknowledgement of the limitation that the system holds is that the handwriting recognition only works for the author's handwriting. A nice implementation feature would be to dynamically train the handwriting of the user in side the application.

From Figure **??** it shows that all that would be needed would be three training examples to help the user to have a good recognition rate. The improvement by making it more generic would be that the application could be extended for a wide range of users.

1.6 Starting again

Although this project has been completed to a degree of standard, there are a few aspects which would be changed if this was to be started again.

When considering the use of the database management systems, a stronger emphasis should have been considered when making the decision when choosing between MySQL and NoSQL. A version of a NoSQL system would have been implemented instead of a relational model, in the hind-sight of the application. This would allow the application to become a generic note-taking application which is not contrained to pure lecture notes for students.

Whilst training the handwriting data it would be imperitive to keep a record of how well the training is doing as it was being performed. This iterative approach at creating a graph would have identified earlier on in the process where the training had stagnated, and should be stopped.

Furthermore, as afforementioned Flask was the wrong decision to use as the core framework for this application. In-light of how the application has grown a better MVC framework, such as Django, although slightly more heavyweight, would have been appropriate. As a result, time would have not been spent on working out trivial tasks.

1.7 Revelence to degree scheme

The author's degree scheme is *Computer Science*. The project has shown a full range of capabilities which satisfy that this project has enhanced and furthered knowledge relating to the subject of Computer Science, as well as enhancing skills already learnt throughout the duration of the author's degree.

The project encorportates many different engineering aspects to make up a programming element.

- It is developed in an agile methodolody process, enforcing good software engineering practices throughout the entire project.
- Design patterns were considered and used throughout the project, prodominently the use of Data access objects and MVC.
- Research work to identify how to binarise a script using computer vison techniques, whilst being aware of the limitation that it can produce.
- Programming was conducted to implement a fully functioning web application, following a code reusability ethos.
- Evaluations and experiements conducted to analyse how well the OCR engine can identify characters.

Overall, there are many aspects of this project which encompass the field of computer science from the research elements, to the analysis and right down to the process followed.

1.8 Overall conclusions

Although alternative implementations and design decisions could have been made it should not be deterred from the application that has been produced.

An application which allows a user to upload their note, tag it with identified meta-data which has been extracted from the user's handwriting and then integrate with this in the calendar is a substantial application. Comprehensive research work was conducted to analyse how Tesseract could have been optimised, and a binarisation script, which works on a variety of images.

This has been conducted with a solid process backing the design, testing and implementation iterative process to show a high quality engineered project.

After substancial work has been completed on this application, the author is delighted with the outcome and believes that the application can be further enchanced with additional features afformentioned, to make it a solid use-case for helping students with their University lecture notes.

Appendices

Appendix A

Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

Apache POI library The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the clients existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [**?**]. The library is released using the Apache License [**?**]. This library was used without modification.

Appendix B

Ethics Submission

This appendix includes a copy of the ethics submission for the project. After you have completed your Ethics submission, you will receive a PDF with a summary of the comments. That document should be embedded in this report, either as images, an embedded PDF or as copied text. The content should also include the Ethics Application Number that you receive.

Appendix C

Code Examples

3.1 Random Number Generator

The Bayes Durham Shuffle ensures that the psuedo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs [?].

```
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0 - EPS)

double ran2(long *idum)
{
  /*---------------------------------------------------*/
  /* Minimum Standard Random Number Generator          */
  /* Taken from Numerical recipies in C                */
  /* Based on Park and Miller with Bays Durham Shuffle */
  /* Coupled Schrage methods for extra periodicity     */
  /* Always call with negative number to initialise    */
  /*---------------------------------------------------*/

  int j;
  long k;
  static long idum2=123456789;
  static long iy=0;
  static long iv[NTAB];
  double temp;

  if (*idum <=0)
  {
    if (-(*idum) < 1)
    {
      *idum = 1;
    }else
    {
      *idum = -(*idum);
    }
    idum2=(*idum);
    for (j=NTAB+7;j>=0;j--)
    {
      k = (*idum)/IQ1;
      *idum = IA1 *(*idum-k*IQ1) - IR1*k;
      if (*idum < 0)
      {
        *idum += IM1;
      }
      if (j < NTAB)
      {
        iv[j] = *idum;
      }
    }
    iy = iv[0];
  }
  k = (*idum)/IQ1;
  *idum = IA1*(*idum-k*IQ1) - IR1*k;
  if (*idum < 0)
  {
    *idum += IM1;
  }
  k = (idum2)/IQ2;
  idum2 = IA2*(idum2-k*IQ2) - IR2*k;
  if (idum2 < 0)
  {
    idum2 += IM2;
  }
  j = iy/NDIV;
  iy=iv[j] - idum2;
  iv[j] = *idum;
  if (iy < 1)
  {
    iy += IMM1;
  }
  if ((temp=AM*iy) > RNMX)
  {
    return RNMX;
  }else
  {
    return temp;
  }
}
```

Annotated Bibliography

[1] "Evernote Tech Blog — The Care and Feeding of Elephants," https://blog.evernote.com/tech/2013/07/18/how-evernotes-image-recognition-works/, 2013, last checked 25th March 2016.

An article explaining how Evernote does character recognition on images

[2] R. Agarwal and D. Umphress, "Extreme Programming for a Single Person Team," in *Proceedings of the 46th Annual Southeast Regional Conference on XX*, ser. ACM-SE 46. New York, NY, USA: ACM, 2008, pp. 82–87. [Online]. Available: http://dx.doi.org/10.1145/1593105.1593127

This paper was useful on how Extreme Programming can be modified to a single person project. It provided thought on the methodology which should be undertaken on the project and how different aspects of Extreme Programming can be used.

[3] Bottle, "Bottle: Python Web Framework Bottle 0.13-dev documentation," http://bottlepy.org/docs/dev/index.html, last checked 22nd April 2016.

The Python framework was used as a case-study of potential frameworks to use for the application. Discussed in the design section, but rejected as a choice.

[4] Django, "The Web framework for perfectionists with deadlines — Django," https://www.djangoproject.com/, last checked 22nd April 2016.

The Python framework was used as a case study, looking at the different frameworks available. It was rejected for it being too large for the project.

[5] M. A. A. Dzulkifli and M. F. F. Mustafar, "The influence of colour on memory performance: a review." *The Malaysian journal of medical sciences : MJMS*, vol. 20, no. 2, pp. 3–9, Mar. 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3743993/

A paper reviewing whether colour helps with memory retention. Used for the analysis and further confirmation in the taxonomy of notes section.

[6] Evernote, "The note-taking space for your life's work — Evernote," https://evernote.com/?var=c, 2016, last checked 17th April 2016.

The Evernote application is an example of the organisational and note-taking application that this project is looking at as a similar system.

[7] Flask, "Welcome — Flask (A Python Microframework)," http://flask.pocoo.org/, last checked 22nd April 2016.

The python framework used as an option. Was used in the design section evaluating the decisions that were made. It was used as the choice of framework.

[8] Google, "Meet Google Keep, Save your thoughts, wherever you are - Keep Google," https://www.google.com/keep/, 2016, last checked 17th April 2016.

Google keep is an organisational and note-taking application, it is used as part of the evaluation and background analysis. It was compared against what the application could do.

[9] R. Gouldsmith, "Ryan Gouldsmith's Blog," https://ryangouldsmith.uk/, 2016, last checked TODO.

A collection of blog posts which explain the progress every week through a review and reflection post.

[10] S. Knerr, L. Personnaz, and G. Dreyfus, "Handwritten digit recognition by neural networks with single-layer training," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 962–968, Nov. 1992. [Online]. Available: http://dx.doi.org/10.1109/72.165597

A paper describing how a Neural network was build to identify handwritten characters on the European database and the U.S. postal service database.

[11] C. Maiden, "An Introduction to Test Driven Development — Code Enigma," https://www.codeenigma.com/community/blog/introduction-test-driven-development, 2013, last checked 17th April 2016.

A blog post giving a detailed description of what Test-driven development includes. Gives supportive detail to discussing that tests can be viewed as documentation.

[12] Microsoft, "Microsoft OneNote — The digital note-taking app for your devices," https://www.onenote.com/, 2016, last checked 13 April 2016.

Used to look at and compare how similar note taking applications structure their application. Used the applicatation to test the user interface and what functionality OneNote offered that may be usedful for the application

[13] ——, "Office LensWindows Apps on Microsoft Store," https://www.microsoft.com/en-gb/store/apps/office-lens/9wzdncrfj3t8, 2016, last checked 17th April 2016.

The Microsoft Lens application which would automatically crop, resize and correctly orientate an image taken at an angle.

[14] ——, "Take handwritten notes in OneNote 2016 for Windows - OneNote," https://support.office.com/en-us/article/Take-handwritten-notes-in-OneNote-2016-for-Windows-0ec88c54-05f3-4cac-b452-9ee62cebbd4c, 2016, last checked 17th April 2016.

An article on OneNote's use of handwriting extraction from an image. Shows simply how to extract text from a given image.

[15] MongoDB, "MongoDB for GIANT Ideas — MongoDB," https://www.mongodb.com/, last checked 22nd April 2016.

The Mongo DB tool used as a comparison for relational database systems and NoSQL ones.

[16] O. Olurinola and O. Tayo, “Colour in learning: Its effect on the retention rate of graduate students,” *Journal of Education and Practice*, vol. 6, no. 14, p. 15, 2015.

Discusses a study which shows that coloured text is better for the memory retention rates, than that of non-coloured text. Used during the taxonomy of notes section.

[17] Oracle, “Overview - The Java EE 6 Tutorial,” https://docs.oracle.com/javaee/6/tutorial/doc/bnaaw.html, last checked 22nd April 2016.

An article which discusses the use of Java as a web application language. It reaffirms the point raised that it is good for performance.

[18] A. Pilon, “Calendar Apps Stats: Google Calendar Named Most Popular — AYTM,” https://aytm.com/blog/daily-survey-results/calendar-apps-survey/, 2015, last checked 13th April 2016.

A survery showing that Google calendar was ranked the most used calendar people use. Added to the analysis stage to justfy why Google calendar was chosen instead of other calendars available.

[19] S. Rakshit and S. Basu, “Recognition of Handwritten Roman Script Using Tesseract Open source OCR Engine,” Mar. 2010. [Online]. Available: http://arxiv.org/abs/1003.5891

The paper presents a case-study into the use of the Tesseract OCR engine. It analyses how to use train the data on handwriting based recognition, drawing conclusions on where it’s useful - as well as it’s downfalls.

[20] Scrum.org, “Resources — Scrum.org - The home of Scrum,” https://www.scrum.org/Resources, 2016, last checked 17th April 2016.

The website for the scrum methodology principles. The website was used to reference the process and methodology which was adapted in the project

[21] T. J. Smoker, C. E. Murphy, and A. K. Rockwell, “Comparing Memory for Handwriting versus Typing,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 53, no. 22, pp. 1744–1747, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1177/154193120905302218

Used to show that there handwriting is still an important part of memory rentention with note taking, compared to digital text

[22] M. G. Software, “Planning Poker: Agile Estimating Made Easy,” https://www.mountaingoatsoftware.com/tools/planning-poker, 2016, last checked 17th April 2016.

Showing the use of planning poker with exactly how it was implemented in the application using the scrum based approach.

[23] Tesseract, “Tesseract Open Source OCR Engine,” https://github.com/tesseract-ocr/tesseract, 2016, last checked 17th April 2016.

The open source optical character recognition engine which will be used in the application to analyse characters on a page.

[24] O. Tezer, “SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems — DigitalOcean,” https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems, last checked 22nd April 2016.

Used as a comparison between what relational management system should be used. Used in the design section for a comparision between the different systems presented and evaluated.

[25] Tiaga, “Taiga.io,” https://taiga.io/, 2016, last checked TODO.

The project management toold which was utilised to help to keep track of the project’s progress throughout the process. Utilised the Scrum tools available that the application gives.

[26] w3Techs, “Usage Statistics and Market Share of JavaScript for Websites, April 2016,” http://w3techs.com/technologies/details/pl-js/all/all, last checked 22nd April 2016.

The website shows a graph of how Javascript has increased its market share on recent web applications. Used as part of the design consideration regarding the use of programming language

[27] M. Webster, “Taxonomy — Definition of Taxonomy by Merriam-Webster,” http://www.merriam-webster.com/dictionary/taxonomy, 2016, last checked 17th April 2016.

A definition of exactly what a teaxonomy is. Clearly labelling it as a classification of a problem.

[28] D. Wells, “CRC Cards,” http://www.extremeprogramming.org/rules/crccards.html, 1999, last checked 17th April 2016.

A description of what CRC cards are and why they’re useful when considering the design of an application. Used as a reference material throughout the process, as well as during the chapter discussing the process.