

MapMyNotes

Report Name	Outline Project Specification
Author (User Id)	Ryan Gouldsmith (ryg1)
Supervisor (User Id)	Hannah Dee (hmd)
Module	CS39440
Degree Scheme	G401 (Computer Science)
Date	February 3, 2016
Revision	0.5
Status	Draft

1 Project description

MapMyNotes aims to produce a web application which will allow the user to upload an image of their handwritten notes and store them in a database. It should also provide the functionality to search for a note and add them to a calendar.

People still handwrite their notes and this often leads to large amounts of paper. This can be cumbersome especially if they try to search through notes which may not be organised. There are modern applications which will allow you to take notes, such as EverNote, but people still like writing handwritten notes.

At a basic level the application will allow the user to upload an image of their handwritten notes. It will use very basic OCR analysis to identify the title of the notes, the lecturer, the module code, the location and the time. Additionally, at the basic level it will interpret the author's handwriting.

From this the application can manually tag the module code for the notes based on suggestions from the OCR analysis, but ultimately the user tags their own notes. The notes metadata (code, location, lecturer and time) will be stored in the database with the note itself. The user can then search for a given module code and find all associated notes, for the specific user. Finally, the user must be able to insert the notes into a calendar, for archival purposes; OAuth with Google Calendar has been considered but more research needs to be conducted.

If there's sufficient time remaining then the application will look to do full OCR recognition on the note and produce a document where the note has been converted to text. There will also be an extraction of diagrams and graphs from the note and the user will be able to drag, rotate etc. An additional further aim is to automatically rotate the image of the note, allowing for notes to be taken at obscure angles and the system will automatically make it perpendicular. Finally, the application could link to other previous notes. These are all long term goals and I will be focusing on the first part mainly.

The methodology that the project will follow will be an adapted Extreme Programming. There needs to be investigatory work on how this can be applied to this project, but during the development stage then CRC cards could be beneficial when thinking about the design stage.

2 Proposed tasks

The following tasks are ones which should be completed on the project:

- **Investigation of how to extract handwriting from images.** This task will involve looking and investigating OCR tools to see how well handwriting data can be interpreted. The tool will need to be trained for the author's handwriting.
- **Investigation into server configuration.** The server will need to be configured with running a web application alongside external libraries such as the OCR tool.
- **Configuration of local work environment.** Configuration of local machines will closely match that of the server, using the same version control, either git or SVN. Configuration of an continuous integration tool to aid project development will be required.
- **Development**
 - **Produce a front end application to input a note.** The front end features must allow a user to upload an image and see the image on the screen. They can add appropriate tags for the module code and they can search for the module code, producing a full list of notes based on the module code.
 - **Back end parsing the images.** The core business logic should conduct basic OCR recognition of text at the top of the notes. The notes can then interact with OpenCV's libraries to extract diagrams from the notes. Finally, the backend module should integrate with a calendar to archive the notes so they can be found again via the date.

- **Produce a list of constraints that the notes must follow.** As notes are varying from person to person, then a constraint should be attached to the notes to ensure the notes follow a similar format. This should be a small set of rules that can be added to the application which represent a formalised structure.
- **Investigation of how to extract diagrams and graphs from images.** There will be research conducted to identify an efficient way to identify images, graphs and diagrams from an image of a note. This could then be implemented into the system.
- **Continuous project meetings and diaries.** The project will consist of weekly meetings with the supervisor. There shall be one group meeting and another individual meeting to show and share progress. A weekly diary will be kept to monitor individual progress on the project and it will be referenced in the final report deliverable.
- **Preparation for the demonstrations.** The project will consist of a mid project demonstration and an end demonstration. The mid project demonstration will need to be prepared to show minimal character recognition of notes and the ability to upload an image. The final project demonstration should show the ability to search for a note by tagging it and archiving it in a calendar.

3 Project deliverables

The following are deliverables which are expected to be completed on the project:

- **The MapMyNotes software.** There should be a web application which at the minimum will take an image and save it to a database and integrate with a calendar. This should be well tested and well structured with appropriate comments where necessary along with providing any build scripts from the server.
- **A collection of user acceptance and model tests.** A series of user acceptance and integration tests should be provided for the front end application. Additionally, back end models should have appropriate logic tested.
- **Story Card, CRC cards, burndown charts and backlog items.** As the author intends to use an Extreme Programming methodology for their project, then providing CRC cards and story cards would be useful for the final report deliverable to show the author's design decisions.
- **OCR training data.** The OCR tool will need to be trained to recognise the author's handwriting. Any training data which aided with this learning stage should be provided.
- **Weekly blogposts regarding progress.** There should be a weekly blog post to aid the author in analysing and reflecting on the week and any obstacles they may have overcome. This will be used and referenced in the final report.
- **Mid-project Demonstration.** There will be a mid project demonstration which should show current progress on the project.
- **The final report.** The report will discuss the work that has been carried out, the process that I have followed any libraries or frameworks which have been used throughout the project. It will discuss the project, the design undertaken and evaluating the end outcome and any changes that would be made.
- **The final demonstration.** There will be a final project demonstration which shows the full extent of the software produced. This should be considered when planning iterations.

Annotated Bibliography

- [1] 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.

- [2] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 6, pp. 910–918, Nov. 1988. [Online]. Available: <http://dx.doi.org/10.1109/34.9112>

This paper was not as useful as it looked. It was soon realised that the application did not need to extract the text from an diagram in the image, instead the image as a whole. This paper may be revisited if there was a need to filter out the text.

- [3] B. M. Gonzalez, "Iris : a solution for executing handwritten code," Master's thesis, University of Agder, 2012. [Online]. Available: <http://brage.bibsys.no/xmlui/handle/11250/137557>

This thesis was a great reference material for handwriting recognition. The used the Tesseract OCR engine to analyse human handwriting and convert the code written on the paper to the actual output. The most interesting part of this thesis was when the author described the training process for handwriting recognition, which shall be considered in this application.

- [4] C. Patel, A. Patel, and D. Patel, "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, vol. 55, no. 10, pp. 50–56, Oct. 2012. [Online]. Available: <http://dx.doi.org/10.5120/8794-2784>

This paper analyses how Tesseract compares to other commercially available OCR software. Although this paper doesn't complete the study on handwriting data it was useful to see the comparisons and the accuracy and success of Tesseract compared to other tools.

- [5] 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>

This paper covers how the Tesseract OCR engine was able to indentify Roman handwriting from scripts. Although this uses an older version of the Tesseract engine, it was useful to know that the Tesseract engine can be used on human handwriting recognition as well as yielding a fairly high accuracy.

- [6] Y. P. Zhou and C. L. Tan, "Hough technique for bar charts detection and recognition in document images," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 2. IEEE, 2000, pp. 605–608 vol.2. [Online]. Available: <http://dx.doi.org/10.1109/icip.2000.899506>

The paper provided a useful and key algorithm in detecting bar charts in images. The paper was interesting as it used a modified Hough transform to group together the collection of lines and bar charts to extract the diagrams from the images. It also showed a way in which I could potentially do this in my application.