

ryan **george** / u1091311

Isaac **Hertweck** / u1081284

ANDREW **JOHNSTON** / u1040508

GREGORY **JONES** / u1023488

**CSC3600**   
ICT Professional Project  
**web based image organiser**

FINAL PROJECT REPORT **supervisor:** a/prof stijn dekeyser

|  |  |  |
| --- | --- | --- |
| VERSION | COMMENTS | DATE |
| 0.1 | Initial template created | 3 Oct 18 |
|  |  |  |
|  |  |  |
|  |  |  |

# VERSION HISTORY

[VERSION HISTORY 1](#_Toc526349024)

[EXECUTIVE SUMMARY 3](#_Toc526349025)

[METHODOLOGY 3](#_Toc526349026)

[METHODOLOGY STATEMENT 3](#_Toc526349027)

[JUSTIFICTIONS 3](#_Toc526349028)

[DISCUSSIONS 3](#_Toc526349029)

[PROJECT PROCESS 4](#_Toc526349030)

[TEAM ORGANISATION 4](#_Toc526349031)

[TEAM STRUCTURE AND ROLES 4](#_Toc526349032)

[COMMUNICATION AND MEETINGS 4](#_Toc526349033)

[DOCUMENTATION 4](#_Toc526349034)

[PROCESS 5](#_Toc526349035)

[PROJECT REPORT 5](#_Toc526349036)

[PROJECT OUTCOME 5](#_Toc526349037)

[COST OF THE PROJECT 5](#_Toc526349038)

[CONTRIBUTION DISTRIBUTION 6](#_Toc526349039)

[CONCLUSION 6](#_Toc526349040)

[REFERENCES 7](#_Toc526349041)

[APPENDIX A: MEETING MINUTES 7](#_Toc526349042)

[APPENDIX B: ACTIVITY LOG SHEETS 7](#_Toc526349043)

***GENERAL POINTS***

***HD: Writing style is consistent throughout report.***

***Writing style is specifically tailored for the intended (non-technical) audience.***

***All paragraphs are appropriately structured so each contains a single topic, and evidence and argument within them are logically ordered.***

***Your paper is free of grammatical and syntactic errors and demonstrates clarity, and sophistication, of expression.***

***In-text references skilfully integrated into the essay in a way that increases its authority.***

***Fully compliant with current Harvard system. Few (if any) errors.***

# EXECUTIVE SUMMARY

Brief description of the project for the purpose, analysis, findings, and recommendations

***HD: The executive summary is an appropriate length (10% under or over inclusive).***

***The summary meets all content requirements: purpose, analysis, findings, and recommendations***

# METHODOLOGY

## 

***HD: The essay addresses all relevant areas in methodology.   
Analysis of methodology employed in the project is in depth and reflective.   
Reflection is well written with lessons learned and knowledge to be carried forward clearly articulated.***

## METHODOLOGY STATEMENT

Notes:

The descriptions about the methodology that you undertook to complete the project. You may use figures to help your descriptions if necessary.

Mark IV Tech implemented an agile software development model to carry out this project. The methodology chosen was scrum.

The user made a list containing must have functionality and desirable functionality. The team leader then selected a system requirement for each member to develop and then implement it. Progress of each backlog task was discussed in daily scrums. The team leader kept the team focused and informed throughout the whole project. Once the component was implemented and tested by the group and was completed and working the team leader would delegate the next lot of tasks.

## JUSTIFICTIONS

Notes:

The discussions about why the specified methodology is selected, and the positive and negative sides of the methodology as revealed in the project.

The scrum methodology was chosen for the agile software development model because it was adaptive to changing user requirements and wishes. Team members lived in different states and some had jobs and therefore the waterfall method was not an option and the software completion date was too short. A negative is due to the lack of documentation there were a couple of occurrences where two team members carried out the same task.

This methodology allowed the team to create a modularized piece of software that could be easily tested, and components could be easily re-developed to a point it met the user’s expectations. A negative throughout the project some members spent too much time on specific tasks.

The team leader kept the team focused on the goal he set each team member throughout the whole project cycle. A minor negative was in some instances feedback took a period of time due to each member being in different states and team work hours weren’t in unison.

## DISCUSSIONS

Notes:

How your team has followed the methodology in the project

How your team will do with the methodology in future in order to achieve a better result of similar projects.

The team had an initial meeting with the user (Stijn) to get a starting point for user requirements. The

team leader broke the project down into task and assigned each team member a task, this document

was loaded onto google docs for all members to see. Each team member completed the component

and loaded it onto GitHub for all members tested and give feedback. The team leader was constantly on

messenger chatting with team members discussing progress and if assistance was required. Once all

members were happy with the component the next set of tasks or system requirements were then

assigned by the team leader.

The results we achieved was exceptional but in future phone and video chat as the main

communication tool would be far more efficient when dealing with complex tasks.

# PROJECT PROCESS

Note: Describe the process that your team followed for this project. Use the headings below to comment on the different aspects of this project.

***HD: The essay addresses all relevant areas in project process.  
Analysis of team problems and (possible) solutions is in depth and reflective.  
Reflection is well written with lessons learned and knowledge to be carried forward clearly articulated.  
The essay is well supported by the documentations (meeting minutes) throughout the project.***

## TEAM ORGANISATION

All forms of team work were organised by utilising Facebook Messenger, Google Docs and via email. Members would discuss availability, proficiencies and weaknesses in terms of coding, report writing and software experience which would provide vital information when organising teamwork.

The team leader Mr Hertweck was an essential figure in ensuring team work was consistently applied throughout the whole project cycle. He would assign two members to a task if there were time constraints or felt a task required skills from two different members that exhibited them.

If a member needed help with a task they would use Messenger to ask a member with the required skills to help. This assistance could have been verbal, providing previous study material or collaborating via GitHub.

A negative to this was when the Messenger chat became cluttered or a member was busy for the day. The member would check the group chat only to find a lot of missed messages that could have been collated into one or two succinct messages. Going forward, the group could be cognisant of this problem and have stricter rules in place.

## TEAM STRUCTURE AND ROLES

Mark IV Tech is made up for 4 team members forming a virtual team. There was a team leader, a front-end department and a back-end department for development. The team leader ran the whole project ensuring all member knew each deadline, assigned tasks to members and kept the team focussed and morale high.

The front-end department was responsible for developing a user interface and the flow of the web application. The back-end department was responsible for ensuring the application talks to the database, saves, retrieves and uses data according to user specifications. For report writing all members were assigned different parts. Then by utilising track changes, another member could check their work and offer suggestions.

This method worked well as early in the process members were selected for either front or back end development based on their strengths in coding. However due to the project requiring more back-end development the workload was not equal during development. This was mitigated by the team leader assigning the front-end members more ancillary tasks and report writing.

## COMMUNICATION AND MEETINGS

Communication throughout the whole project was exceptional because of the democratic management style adopted. All members were encouraged to communicate and use multiple platforms to achieve this. However, in some instances, due to the complexity of the issue or task the platforms used were not adequate. The length of time to discuss an issue or explain a view took longer than it should have. Finally, availability of each member was mixed as members had to balance work-life and family along with development.

For future collaboration, phone calls or regular video conferencing could have provided live discussion and would have been far easier to discuss complex issues and tasks. As half of the team were in a different state, face-to-face meetings were not possible. In future it would have been beneficial to be located within travelling distance.

All team meeting was organised 3 to 4 day in advance and the video conference would be run on the zoom platform. These meetings were extremely successful because every member was able to easily discuss or bring up any difficulties they were having. Furthermore, it was far easier to discuss project direction and tasks requiring help through verbal communication than instant messaging.

To keep meetings on topic, an agenda could have been loaded onto Google Docs where each member could have added to. That would have given the team an approximation of meeting length and ensure all required topics were discussed. The team meetings were short and concise due to time constraints. For these, meeting minutes were taken and scribed for future reference.

## DOCUMENTATION

The group benefited from the platforms used supporting full history of conversations and changes. This allowed members to read through the chat history before re-asking similar questions. GitHub also allowed members to see committed changes and the version history. Thus allowing them to know exactly what stage of development it was up to.

All meeting minutes and summaries of fortnightly conversations have been attached to this document as Appendix A.

## PROCESS

Notes:

• Describe the process followed by your team during the project.

• What difficulties did you encounter and how they were or were not over-come?

• What activities or strategies did you employee that you thought were successful and should be done on future projects?

# PROJECT REPORT

## PROJECT OUTCOME

Mark IV Tech was recently assigned the task of creating a web-based image organiser in the wake of Google Picasa, a similar service that was discontinued in March 2016. A number of software requirements were specified by the client in the original project plan. These requirements included but were not limited to:

* Read/Write SQLite database.
* Ability to set a root directory according to user selection.
* Extract all metadata from user selected root directory.
* Save edited metadata back to the image.
* Default view to display all thumbnail images within the repository.
* Sort thumbnail images according to date (newest to oldest).
* Perform a simple single field search for images according to user input.
* Responsive GUI.
* User-friendly interface.
* Maintain data integrity.
* Operate across multiple platforms and devices.

Using an agile methodology for our development cycle, we are happy to report that we have successfully fulfilled the requirements of the client, as well as including some additional features throughout the development cycle. We believe that the final product delivered is both high in quality and user-friendly. Some of the additional features included are:

* Re-sort images by date (oldest to newest).
* Perform complex multiple field search for images according to user input.
* Allow user to update fields of image metadata.
* File system metadata synchronised with database data for each user request.
* Dynamic scanning of image directories.
* Additional database information.

The web-based image organiser was designed with usability and functionality in mind. The front-end design utilises the Bootstrap framework in order to present a clear and concise webpage to the user. SQLite 3 was chosen as the preferred method of storing and querying database information, due to the nature of the project. This allows the user to run the software solely using an Apache server.

It is recommended that users take advantage of the “XAMPP” software package when using the web-based image organiser, as it is available on multiple platforms and was primarily used by the development team when creating the software.

### LIMITATIONS

The software we have created is best used in a non-commercial environment, as there are limitations to the size of each image directory. Images must be stored in JPEG/JPG format, otherwise they will not be scanned and updated by the software. Using larger image directories may take the software significantly longer to scan through, depending on the size of each directory.

### SCANNING IMAGES

Users can select multiple directories to scan images from, and can later remove the stored database, effectively removing all image directories used by the software. If entering a drive letter to locate your image directories, be sure to include the colon after the drive letter e.g. “C:”. Please note that the “Browse Images” page, simple search, and “Advanced Search” page will not function properly until at least one directory has been added to the software.

### BROWSING IMAGES

Users can view their image library in this section, and can modify image metadata by clicking on an image and changing information stored in the metadata fields. The default image library view is sorted by year (newest-oldest), but can be changed to sort between (newest-oldest) or (oldest-newest) by using the “Sort Images” button at the top of the page. Any metadata changed within the information fields must be saved first in order to be used by the search functions.

### DATABASE INFORMATION

This section of the software allows the user to view information about the database, such as the path of their image directories, the number of images are stored in the library, and the size of the database. Users can also delete their current database on this page, which will remove all photos from the image library on the software. If multiple directories are stored, they can also be individually removed by clicking the “Remove” button next to the corresponding directory path.

### SIMPLE SEARCH FUNCTION

Users can search for metadata keywords stored within the image library by using the simple search function located on the top-right of each page (*Note: simple search function is not available on the advanced search page*). Parts of keywords will also return results e.g. searching “fire” returns results with stored metadata “firetruck”.

### ADVANCED SEARCH PAGE

Users can search specific metadata fields using this page. Please note that if multiple fields are used, any results returned will be inclusive, not exclusive. Dynamic dropdown fields are used to select specific camera manufacturers and/or camera models. If a manufacturer is chosen first, the camera model field will be updated with the respective models for that manufacturer. If a model is chosen first, the manufacturers field will update to that model’s respective maker. These fields can be reset by using the reset button located directly below the dropdown fields.

### OTHER FEATURES

If individual images are removed from a root directory (outside of the software), they will also no longer show up on the image library. On the contrary, if an image is added to a root directory, they will also be added to the image library. If no directories have been selected, the user will be prompted to add a directory and will be taken to the “Scan Images” page where they can add one.

### FUTURE UPDATES

Mark IV Tech uses GitHub in order to collaborate on their projects. Future updates to the web-based image organiser can be found online at <https://github.com/Gregory1999/CSC3600>. Most of the source code has been commented on and structured properly so that it can be built upon and easily understood by other developers.

## COST OF THE PROJECT

(Note: will update figures a couple of days before due date)This project has had a total project expenditure of $17,445.79. The initial plan anticipated an outlay of $70,199.75, this total however was caused by an uncaught error in which the total labour costs were incorrectly multiplied by four; the corrected estimated expenditure was $20,129.00. The project has therefore underspent anticipated (corrected) costs by $2,683.21. The cost of each project is broken down and displayed in the below tables. The Labour table sources its data from the task log summary shown in appendix XX.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Phase** | **Total Hours(4)** | **Cost (Business Analyst Rate(1))** | **Cost (Programmer Rate(2))** | **Cost (Project Manager Rate(3))** | **Total Phase Cost** |
| **1** | **Initialising** | 27.75 | $1,045.98 | $0.00 | $505.81 | $3,403.20 |
| **2** | **Discover and understand the details of the problem** | 31.5 | $1,191.93 | $0.00 | $566.51 | $4,537.60 |
| **3** | **Create the project plan.** | 41.5 | $1,654.10 | $0.00 | $606.98 | $7,260.16 |
| **4** | **Design Components** | 70 | $0.00 | $1,855.00 | $1,375.81 | $1,680.00 |
| **5** | **Build all the program components, integrate and test.** | 103.5 | $0.00 | $3,325.00 | $687.91 | $25,200.00 |
| **6** | **Monitoring and Controlling** | 17.5 | $316.23 | $0.00 | $890.23 | $4,764.48 |
| **7** | **Deploy** | 0.5 | $24.33 | $0.00 | $0.00 | $8,394.56 |
| **Totals** | | **292.25** | **$4,232.55** | **$5,180.00** | **$4,633.24** | **$14,045.79** |
| **Notes:** | | | | | | |
| 1. Business Analyst Hourly Rate-$48.65 (PayScale 2018). | | | | | | |
| 2. Programmer Hourly Rate-$35(PayScale 2018). | | | | | | |
| 3. Project Manager Hourly Rate-$80.93(PayScale 2018). | | | | | | |
| 4. Phase hours sourced from Task Log Summary, refer Appendix XX. | | | | | | |

**Table XX: Labour- Time and Cost Estimation**

|  |  |  |
| --- | --- | --- |
|  | Resources | Cost |
| Software | Virtualisation | $0.00 |
| GitHub | $0.00 |
| PHP/MySQL/AJAX | $0.00 |
| Hardware | Desktop PCs x 4 | $2,400.00 |
| Internet Access (Project Duration) | $1,000.00 |
| Total Cost | | **$3,400.00** |

**Table XX: Resource Cost Estimation**

|  |  |
| --- | --- |
| Title | Cost |
| Total Resource Cost | $3,400.00 |
| Total Labour Cost | $14,045.79 |
| Total Project Cost | $17,445.79 |

**Table XX: Total Project Expenditure**

|  |  |
| --- | --- |
| Title | Cost |
| Planned Expenditure (Corrected) | $20,129.00 |
| Actual Expenditure | $17,445.79 |
| Expenditure Delta | -$2,683.21 |

**Table XX: Project Expenditure Comparison**

The below graph displays a comparison of the planned lobour hours (in red) and the actual hours carried out during the project (shown in blue). From the graph it is shown that there are two phases which where inaccurately forecasted. The project team only required 43% of the anticpated programing and testing hours and exceeded the design phase hours by almost 6 times.

**Figure XX: Planned vs Actual Labour Comparison**

The above graph highlights the first costings lesson learnt, which is to not underestimate the time required to carry out the projects design, especially when working within a team. Most university assignments have been individual activities, so less time is required to create and communicate design details, with more time spent coding. It was noted through this project that extra time is required, when working in a team environment, to discuss and specify the proposed design. In future projects this team would actively proportion less time to programming and allocate more labour resources to develop designs.

The final costings lesson learnt during this project was to be more diligent with the financial calculations. The accounting error within the initial plan was caused by miscommunications during the early phases of the project, before communication paths and norms had matured, and was compounded by the dispersed structure of the project team. The error was then not identified as the team had not had experience in software projects so had little idea of the expected scale of the expenditure. To mitigate this error in future projects the team would employ additional Independent checks and validation for all critical financial calculations.

# CONTRIBUTION DISTRIBUTION

Notes:

* Fill the table with approximate percentage of effort each team member contributed to each part of the project. (Team member names go along the top row.)
* Tasks include all activities in the project, such as research, discussions, documentation, meetings, and implementation.
* Comment on how you distributed the workload within the team.
* Comment on how well your distribution of tasks worked

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MAIN  TASKS | RYAN  GEORGE | ISAAC  HERTWECK | ANDREW JOHNSTON | GREGORY  JONES |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| TOTAL |  |  |  |  |

# CONCLUSION

Notes:

• Summarize the information presented in the document

• Summarize your team's project experience.

• What did you learn?

• How do you plan to run your next project?

# REFERENCES

Notes:

• List the references that are mentioned in the document or may support the information in the document.

# APPENDIX A: MEETING MINUTES

Note: Attach all meeting minutes taken throughout the project here

# APPENDIX B: ACTIVITY LOG SHEETS

Note: Attach all activity log sheets recorded throughout the project here