



*Division of Computing Science and Mathematics
Faculty of Natural Sciences
University of Stirling*

Visualising Learning Analytics Data at the University of Stirling

Victoria Szymanska

**Dissertation submitted in partial fulfilment for the degree of
Master of Science in Information Technology**

September 2018

Abstract

The project described in this dissertation was conceived by the University of Stirling in order to explore the capabilities of using Tableau Desktop to visualise data about student activity from Canvas. This project was inspired by the activities taking place in the Scottish sector to advance the field of learning analytics.

A problem was identified when it was discovered that Canvas provides two data sets, namely Canvas Data and Live Events, which are unable to be joined together in Tableau. The real problem to solve in this project was then to figure out how to combine the two datasets in such a way that they become usable in conjunction with each other. Canvas Data details everything about the current state of Canvas, including the likes of student grades, current enrolments and details about assignment submissions. Live Events, on the other hand, contain real-time information about events occurring in Canvas – i.e. every click.

The problem lied in the fact that Canvas Data was stored in neatly structured tables in a MySQL database, but the bulk of the necessary Live Events data were supplied in JSON format and stored in one column of a Live Events table in the same MySQL database. When trying to interface with this data, Tableau was simply unable to read it.

The tasks for this project became two-fold: first, restructure the Live Events data in to a format readable by Tableau, and second, identify relationships in the data in order to make visualisations according to user requirements.

Canvas offers some analytics capabilities built in to the service called Canvas Analytics, which provides information about student performance and activity within Canvas. Although the analytics presented in this service are useful, they cannot be used to provide an overview of how a student is performing throughout their degree course as the analytics are only available for each individual module. Staff at the university identified the need for a reporting tool that could unify the data and provide a more holistic view of the student experience.

This project was successful in transforming Live Events into a usable format by using a combination of SQL and PHP in a script that retrieved the data, de-constructed it and re-uploaded it into a new MySQL table. The project was also successful in meeting user requirements and creating Tableau dashboards with visualisations from both sets of data combined. Overall, the project was considered a success.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following:

- The SQL commands used to populate the Canvas Data fields described in Chapter 4 were written by Simon Booth, Infrastructure Team Leader at the University of Stirling.
- The original Tableau workbook displaying student grades was created by Larry Kirk, Project Manager at the University of Stirling. This work was expanded on by me.

Signature

Date

Acknowledgements

I would like to thank Larry Kirk and Simon Booth of the University of Stirling for trusting me with this project and for giving me the freedom to experiment in such an exciting field.

To my partner, Tam, I am eternally grateful for the support (particularly in the last few hours before this was due). I could not have made it without you.

Table of Contents

Abstract	i
Attestation.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Figures.....	vi
1 Introduction	1
1.1 Background and Context	1
1.2 Scope and Objectives	2
1.3 Achievements.....	3
1.4 Overview of Dissertation.....	3
2 State-of-The-Art	4
2.1 Canvas Analytics	4
2.2 Canvas Data Portal.....	6
2.3 Tableau	6
2.4 EU General Data Protection Regulation (GDPR).....	8
3 Requirements Engineering.....	9
3.1 First Iteration	9
3.1.1 Initial User Requirements.....	9
3.1.2 Tableau User Requirements	9
3.1.3 Use Cases	10
3.2 Second Iteration	10
3.2.1 User Requirements.....	10
3.2.2 Use Cases	11
3.3 Third Iteration.....	12
3.3.1 User Requirements.....	12
3.3.2 Use Cases	12
3.4 Non-functional Requirements.....	13
3.4.1 Dashboard Requirements	13
3.4.2 Database Design Requirements	13
3.5 Requirement Analysis.....	13
3.6 Achievable Goals	15
4 Data Structuring and Analysis	16
4.1 Canvas Data	16
4.1.1 Retrieval and Processing	16
4.1.2 Data Model.....	16
4.1.3 Data Exploration.....	17
4.2 Live Events Data	21
4.2.1 Retrieval	21
4.2.2 Data Selection and Processing	23
4.2.3 Selecting an Event Type.....	23
4.2.4 Selecting the Fields	25
4.2.5 Creating and Populating the New Live Events Table	27
4.2.6 Redesigned Live Events Data Model	28
4.2.7 Data Exploration.....	30
5 Visualising with Tableau.....	32
5.1 Initial Investigation	32
5.2 Dashboards.....	34
5.2.1 Assignments and Grades Dashboard	34
5.2.2 Access and Activity Dashboard	35

5.2.3	Frequency of Clicks on Module Content Dashboard.....	37
5.2.4	Feedback Access Dashboard.	38
6	Evaluation.....	39
6.1	Evaluation Against Set Goals	39
6.2	Evaluation of Dashboards.....	40
6.3	Evaluation of the Data Design	41
7	Conclusions	43
	References.....	44
	Appendix 1	46

List of Figures

Figure 1. Canvas Analytics at module level.....	4
Figure 2. Canvas Analytics at student level.....	5
Figure 3. Canvas Data and Live Events; two datasets to be joined.....	6
Figure 4. An example of visualisations created using Tableau[14].	7
Figure 5. Tableau Dashboard embedded in Canvas.....	7
Figure 6. Use Case Diagram for the First Iteration.....	10
Figure 7. Use Case Diagram for the Second Iteration.	11
Figure 8. Use Case Diagram for the Third Iteration.	12
Figure 9. Requirements and Implication for the Appearance of the Dashboards.....	14
Figure 10. Requirements and Implications for the Structure of the Data.	14
Figure 11. Diagram representing the star schema on the Enrollment_fact table.....	17
Figure 12. User Requirements Matched with Canvas Data.	18
Figure 13. Results of SQL query selecting student grades.....	19
Figure 14. Results of the SQL query selecting for data about assignments submitted on time.	20
Figure 15. Screenshot from Tableau Displaying the Original Live Events Table.....	21
Figure 16. A sample of the Live Events <i>Body</i> field for an “Asset_accessed” event.....	22
Figure 17. An example of nested JSON data describing the “actor” in an event.....	23
Figure 18. The count of distinct records per <i>event_name</i> ordered from most to least.	24
Figure 19. Selected Live Events attributes and their descriptions.....	26
Figure 20. Location of JSON attributes stored in the PHP variable \$obj.....	27
Figure 21. Diagram representing the Relationships between Live_Events2 and Canvas Data ...	28
Figure 22. Relationships between <i>object_extensions_entity_id</i> and Canvas Data.	29
Figure 23. Results of SQL query selecting for number of events by day.....	30
Figure 24. Results of SQL query selecting for students, file names and the times they were clicked on by the student.	31
Figure 25. Canvas Data tables joined in Tableau.....	32
Figure 26. Joins and join clauses depicted in Tableau.....	32
Figure 27. Field options Menu displayed in Tableau.....	32
Figure 28. Example of a Calculated Field.	33
Figure 29. Example workbook displaying student grades.....	33
Figure 30. Student Grade Review Example Dashboard.	33
Figure 31. Joins used for the Assignments and Grades Dashboard.....	34
Figure 32. Calculated field used for Assignment Submission Statuses.....	34
Figure 33. Assignments and Grades Dashboard.	35
Figure 34. Joins used for the Access and Activity Dashboard.....	36
Figure 35. Access and Activity Dashboard.	36
Figure 36. Joins used for the Frequency of Clicks on Module Content Dashboard.	37
Figure 37. Frequency of Clicks on Module Content.....	37
Figure 38. Joins used for the Feedback Access Dashboard.....	38
Figure 39. Feedback Access Dashboard.....	38

1 Introduction

1.1 Background and Context

The last few years have seen a rise in the Big Data culture that has so deeply engrained itself into our daily lives. Data analytics has been a popular topic of discussion amid major scandals such as Cambridge Analytica[1] where millions of Facebook users' profile data were compromised in a data breach making concerns over data security grow. Even so, business intelligence tools and machine learning algorithms are present in almost every facet of our lives. The music suggested to us on streaming services is carefully curated through an analysis of our data[2] and the popularity of AI-enabled home assistants such as Amazon's Alexa is on the rise[3].

Although there is a feeling that such large-scale analysis of our data can be nefarious, there are institutions that are looking to use data for the purposes of good, namely: universities and colleges in the education sector. Their aim is to use a particular branch of analytics called *learning analytics* to gain insight about students to better assist them through their educational careers.

Learning analytics is defined as the "measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs"[4]. It is in this branch of analytics that the project outlined in this dissertation has its roots.

There has been a growing interest in learning analytics at Scottish universities. Quality Assurance Agency (QAA) Scotland is the quality body for Higher Education in the Scottish context[5][1]. One main area of work for QAA Scotland is the Enhancement Themes, which are coordinated programmes of activity involving the whole Scottish Higher Education sector. Through the Themes, the sector collaborates on improving strategy, policy and practice [6][7]. The current Enhancement Theme (2017-2020) is titled "Evidence for Enhancement: Improving the Student Experience" and aims to investigate the evidence that is used to identify and evaluate the positive and negative aspects of the student journey.

Learning analytics is one of the main points of interest for the current Enhancement Theme with both sector-wide and clustered activities taking place to explore the use and development of learning analytics in Higher Education[8]. The University of Stirling is involved in a collaborative cluster and has hosted two research interns to further scope out the landscape of learning analytics in the Scottish Higher Education sector[9]. Furthermore, the university also has a Learning Analytics Steering Group in place that oversees the development of any work done in this field.

Wider work in the UK Higher Education sector is also being carried out in the field of learning analytics. Jisc (a digital solutions provider for UK education and research) is collaborating with commercial suppliers to build a learning analytics service for the sector[10]. Over 50 UK universities are involved in piloting the first implementation of the service and the University of Stirling will be taking part in this pilot. The Jisc service is based on a series of apps and technologies that record data about students to use for reporting, one of which is an attendance monitoring app. Because learning analytics is concerned with learning and all contexts it takes place in, the series of measures that can be used for analysis are vast and range from student record systems to library data. Interest in the Jisc service so broadly across higher education reflects a demand for analytics solutions among an increasing interest in data analytics.

With a view of the current landscape, the University of Stirling are considering the possible solutions that could provide better insights into their student population. One useful source of student data is the Canvas Learning Management System(LMS), created by Instructure, which is used by the university to make course content available online. Most of the courses run by the university are structured in such a way that students must make use of Canvas in order to

retrieve materials and view information about their modules, submit assignments and view their grades. By making students engage with Canvas in this way, a vast amount of data is accumulated about student activity and performance. This makes Canvas an ideal candidate for data analysis about student learning.

Canvas Analytics is a service offered by Canvas that allows instructors on courses view the activity and performance data of their students. Although this can be a useful tool, the university finds it limiting as it only offers insight into how individual students are engaging with specific modules, not an overall view of their data.

In order to gain a better understanding of how students are performing overall, the university has subscribed to the Canvas Data Portal which makes all of their Canvas Data available to use for reporting and querying. This has opened up an opportunity to view student data in a broader sense than just within the context of their individual modules.

The university also subscribed to a new feature called Live Events, which details every event taking place within the Canvas LMS. This data contains a better insight into activity of students as it details every click, upload or interaction they have within the LMS.

The university is interested in using the two datasets to gain insight into the activity of their students. A license for the business intelligence tool, Tableau, was purchased to use for exploring the data which is how this project was born.

1.2 Scope and Objectives

The aim of this dissertation project is to explore how Tableau can be used to visualise data from Canvas to help make data about students available to staff.

The reports and visualisations must be of a broader scope than what is currently offered by Canvas Analytics and must be able to show information about learners in the context of their whole degree programme, not just information specific to a given module.

From the initial consultations, two key issues were identified:

1. Live Events data is a new feature offered by Canvas and is unexplored territory. The data is being supplied but has not yet been interpreted is poorly documented.
2. Tableau is currently unable to interact with Live Events data as it is being supplied and stored in an incompatible format.

With the above in mind, the following objectives were identified:

- Gain an understanding of the Live Events: identify the structure and format of the data; understand what the values represent in reference to what is seen in the LMS.
- Identify the links between Live Events and existing Canvas data so that foreign keys and table joins can be implemented.
- Format Live Events in such a way that it can be queried in conjunction with existing Canvas data using Tableau.
- Identify the reports and visualisations that it is possible to generate in Tableau using the given data with a view to satisfy user requirements.
- Identify different views of the data so that a student dashboard and a staff dashboard can be created displaying different aspects of the data that is relevant to them.
- Investigate the possibility of embedding Tableau dashboards into Canvas or a web page so that staff and students can view it.

The project concerns itself primarily with descriptive analytics as opposed to predictive analytics. It is out with the scope of this project to create a predictive model or to display predictions about students. It is only the intention to create a view of what is currently happening, not what is predicted to happen in future.

The project is also solely focused on data that can be retrieved from Canvas and does not include other measures such as data from SITS (the student records system) or any other source.

The scope of this project is to explore the possibilities with Tableau with the understanding that the resulting visualisations and reports created for this project might not be used.

1.3 Achievements

The below outlines the key achievements of this project:

- The relationship between Live Events and Canvas Data was identified.
- Live Events were restructured from a NoSQL format into a MySQL table, enabling it to be used in conjunction with Canvas Data in Tableau.
- Tableau was used to create dashboards displaying activity and performance data about students.

The below details what was unable to be achieved:

It was made clear through the process of doing this dissertation project that it is not possible to embed Tableau dashboards on the web unless the university invests in Tableau Server. As this was an exploratory exercise into the possibilities of Tableau it was not possible to make such an investment for the purposes of this project.

1.4 Overview of Dissertation

The dissertation will outline the steps taken to reach the objectives set out for the project. Below is a description of what each chapter details:

Chapter 1 – The background and context of the project will be outlined, objectives set, and achievements listed.

Chapter 2 – The current state-of-the-art technologies and solutions used to solve similar problems in the real world will be discussed.

Chapter 3 – User requirements and their implications for design will be detailed and achievable goals will be set as a basis for evaluation.

Chapter 4 – The process of retrieving, structuring and exploring the data to meet user requirements will be discussed.

Chapter 5 – The process of constructing queries and creating visualisations with Tableau will be described, with a focus on how user testing has influenced the process.

Chapter 6 – Evaluation of the project results will be discussed, making use of the goals set out in chapter 5 and the eight golden rules of UX design as metrics for evaluation.

Chapter 7 – A summary of the project will be provided and examined, conclusions drawn and future work suggested.

2 State-of-The-Art

2.1 Canvas Analytics

Canvas Analytics offers institutions insights into how well the users enrolled on their courses are performing. It is a helpful tool for comparative analytics within each module, as it offers an aggregated view of data for the module as well as individual student statistics detailing their performance within the module.

The below screenshot is an example of the types of module analytics that can be seen in Canvas Analytics. By aggregating all of the student data in each course, spikes and dips in overall module activity can be seen in the “Activity by Date” graph, submission statuses for each assignment are colour-coded in the “Submissions” graph, and the range, median and 25-75 percentile of student grades are displayed in the “Grades” section.

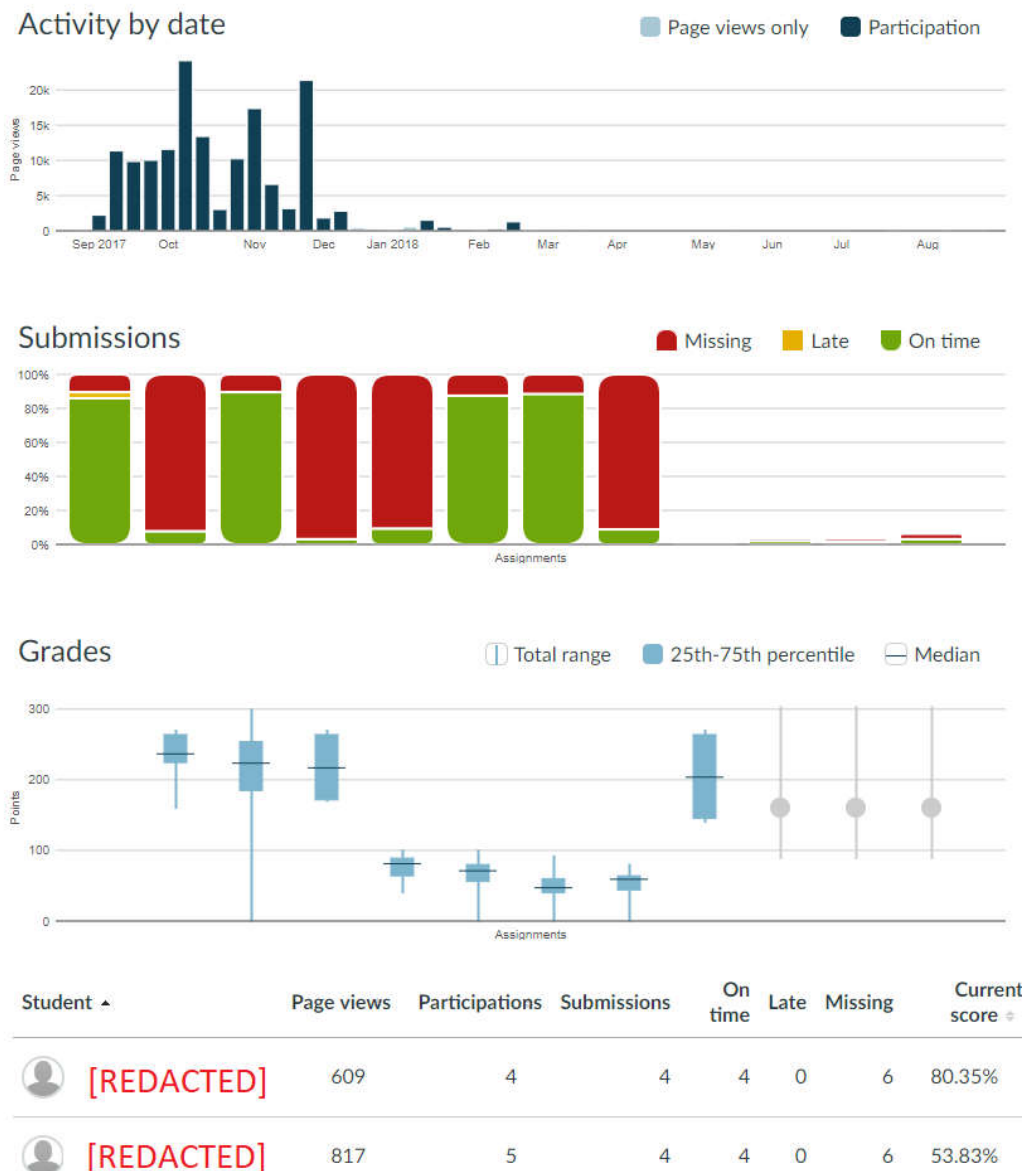


Figure 1. Canvas Analytics at module level.

Similar data can be found when looking at Canvas Analytics for individual students as seen in the example below:

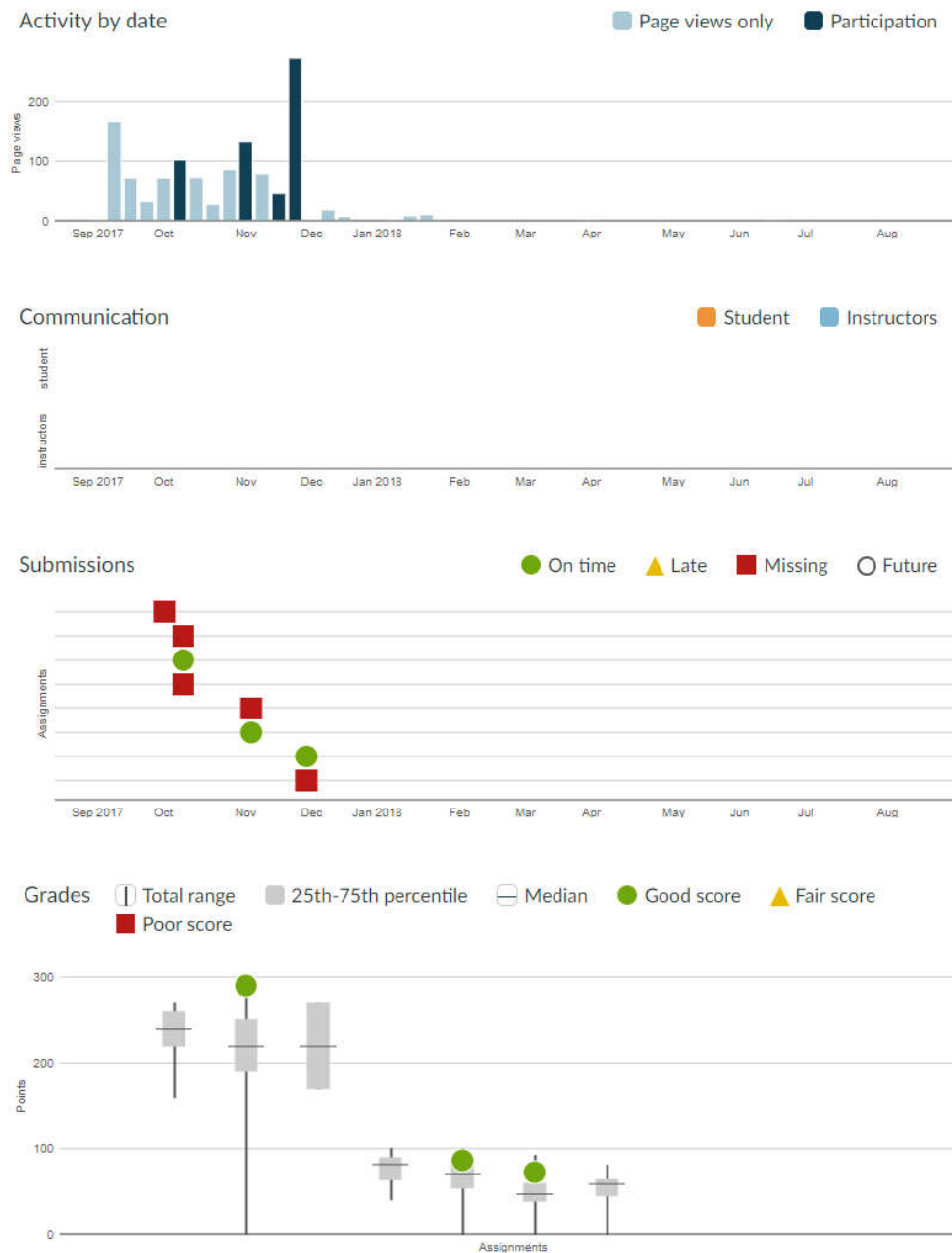


Figure 2. Canvas Analytics at student level.

It is important to note that Canvas Analytics provides comparative data, meaning that students are compared to the peers on their course. In the grades example above it can be seen that the student performs quite well on assignments that are handed in, as indicated by the green dots on the graph. Grey bars are also used to indicate the 25th-27th percentile of grades in the class. Such comparisons may be useful to module co-ordinators or teaching staff that want to see exactly how one student is comparing to the rest of their class.

The layout and proportions of graphs presented in Canvas Analytics are quite simple, with language that is easy to understand, and legends provided for all of the symbols used.

The aim is to emulate the simplicity of these dashboards in this project by providing data pertaining to a student's overall performance.

2.2 Canvas Data Portal

The university is subscribed to the Canvas Data Portal which is a service that allows institutions to access their data from Canvas.

Canvas make use of Amazon Web Services for hosting their data and make everything available to their universities via Redshift or downloadable through the Canvas Data Portal.

Canvas Data is extensive and details the current state of university activity in Canvas. It holds details about the modules that students are enrolled, the grades they have received on assignments, details about module pages, files uploaded by teaching staff, user interface navigation information and even external tools activated from within Canvas.

Over all, there are 88 tables each containing thousands of records. All of the data above are referred to collectively as Canvas Data and are made available as a data dump once every 24 hours.

Live Events is an addition to Canvas Data that was put into production in February 2018[11] although it was available as an experimental feature prior to this and can be traced back to 2017[12]. Live Events records real time activity within Canvas including user logins, file uploads, account creations, and page views. It captures anything that could be considered a 'click' or event anywhere within Canvas. The Amazon Simple Queue Service (SQS) is used to provide universities with Live Events in real-time.

Both Canvas Data and Live Events are collected, stored, and uploaded to a MySQL database hosted by the University. Although the data is stored together in the MySQL database, it was identified that the way in which Live Events is stored makes it incompatible for use in conjunction with Canvas Data.

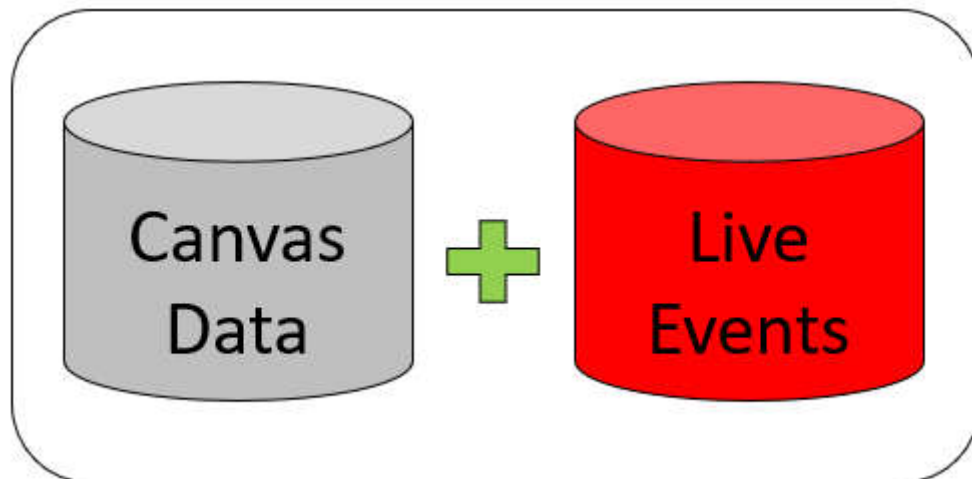


Figure 3. Canvas Data and Live Events; two datasets to be joined.

2.3 Tableau

Tableau Desktop is a business intelligence tool that allows users to create visualisations and run analysis on large datasets [13]. This software is intended to be licensed for use by individuals, however, the Tableau company also create a variety of products that can be used at a larger scale for businesses such as Tableau Online and Tableau Server. These two tools enable companies to host their data and visualisations online within the Tableau cloud.

Tableau Desktop can connect to a variety of relational and cloud-based databases and enables users to query their data by using an interactive graphical user interface (GUI). The software provides quick and easy methods to create graphs and charts using drag-and-drop functionality. This makes it easy for users to interact with their data and analyse it at scale without the need to have any coding background in database query languages such as SQL.

The product used in this project is Tableau Desktop, which will be referred to as ‘Tableau’ throughout.

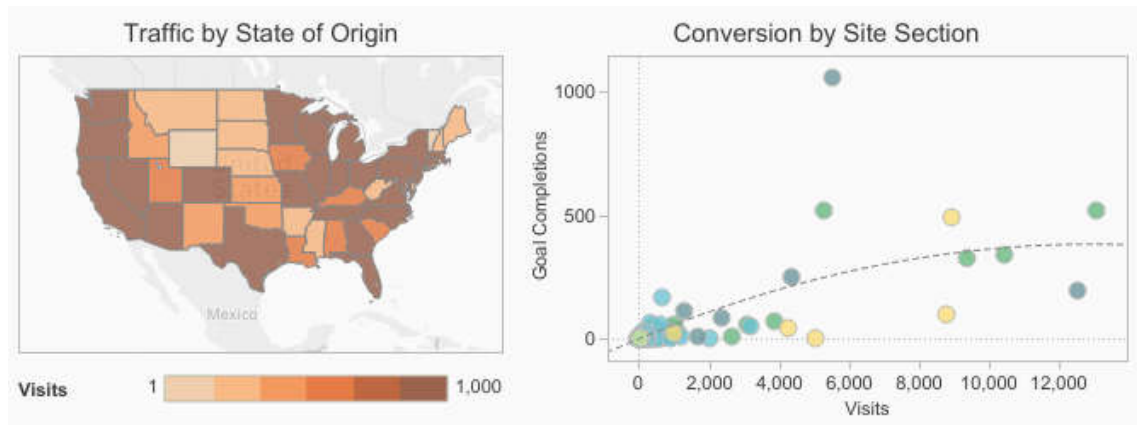


Figure 4. An example of visualisations created using Tableau[14].

Tableau is also used by other academic institutions to visualise information about their data. It has even been used in conjunction with Canvas Data, and many case studies have been published in documents on the Canvas Community pages (the online forum for Canvas users)[15].

The below image shows an example of a dashboard from one such case study. The dashboard details different information about student enrolments using Canvas data. It can be seen that the dashboard is embedded in Canvas as indicated by the navigational menu on the left.

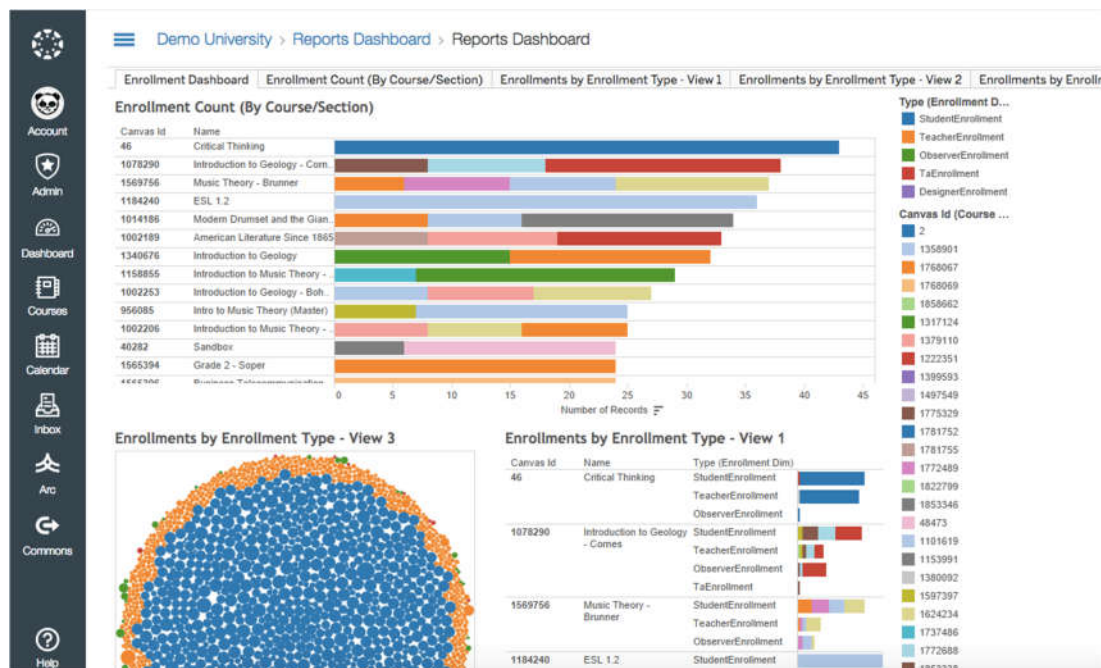


Figure 5. Tableau Dashboard embedded in Canvas.

The dashboards in these case studies are often shared so that more institutions can make use of them. This possibility was investigated for the purposes of this project, however, due to most institutions using Amazon Redshift to store their Canvas Data, the process of switching this data source for a MySQL database would make the workbook incompatible.

Similar business intelligence tools include the likes of Microsoft Power BI and Azure Suite. These were not considered for use in this project due to the limitations set indicating that Tableau must be used.

2.4 EU General Data Protection Regulation (GDPR)

All data processed in this project was done in line with the principles and regulations outlined by GDPR[16]. For the purposes of this dissertation, sometimes student data must be displayed in examples or figures, in which case, any features that would make the student identifiable will be redacted to protect their data. The only cases in which data will be displayed openly is if it is my own.

3 Requirements Engineering

This chapter discusses how the requirements of the users were identified and how this shaped the direction of the project. A spiral prototyping software development approach was used throughout the project and as such, there was an opportunity for new requirements to be introduced in each iteration of the development cycle. The requirements are therefore described below in iterations. The goals outlined in this chapter are used as a basis against which to evaluate the project.

3.1 First Iteration

Initial user requirements were elicited in discussions among members of the Learning Analytics Steering Group and academic colleagues engaged in improving the student experience. The requirements were gathered by the Project Manager for Learning and Teaching at Information Services and presented for use in this project.

The initial discussions took into consideration the current functionality provided by Canvas Analytics and how these may be developed to provide a more holistic view of the student experience.

3.1.1 Initial User Requirements

The below list of reporting requirements was compiled from feedback during initial discussions. It details the information that staff would find useful from a learning analytics platform:

- Data about grades for all modules undertaken by a student, by category of assignment.
- Information on activity recorded within Canvas (similar to the “Activity by Date” provided by Canvas Analytics) across all modules.
- Information on extensions provided on assignments.
- Statistics on median module scores across each division, within faculties. This can be split into Undergraduate/Postgraduate and build up over semesters as required.
- Data about when students are accessing specific module content such as lecture slides or resource material – functionality that is not currently provided by Canvas Analytics.
- Data about students should be visible to themselves.

3.1.2 Tableau User Requirements

With the view that Tableau would be used to create reports and visualisations to be embedded into a set of dashboards, the above initial requirements were considered and a set of additional user requirements specific to Tableau were agreed as below:

- Individual dashboards must be created for two different user types: students and staff.
- The dashboards must contain information that is relevant to each user group.
- Students should only be allowed to view data about themselves if the reports are at an individual level.
- Tableau should be used to expand on the information presented in Canvas Analytics by aggregating data about student activity and grades across their whole degree programme, as opposed to the modular view currently provided.

- The project should investigate the ability to embed the Tableau dashboards on a web page or within Canvas to make them accessible to the intended users via desktop.

3.1.3 Use Cases

The above requirements were used to generate use cases as detailed in the diagram below.

Three of the use cases were identified as applicable to both students and staff members, as these pertain to individual student data. The only restriction is that students should only see information about themselves.

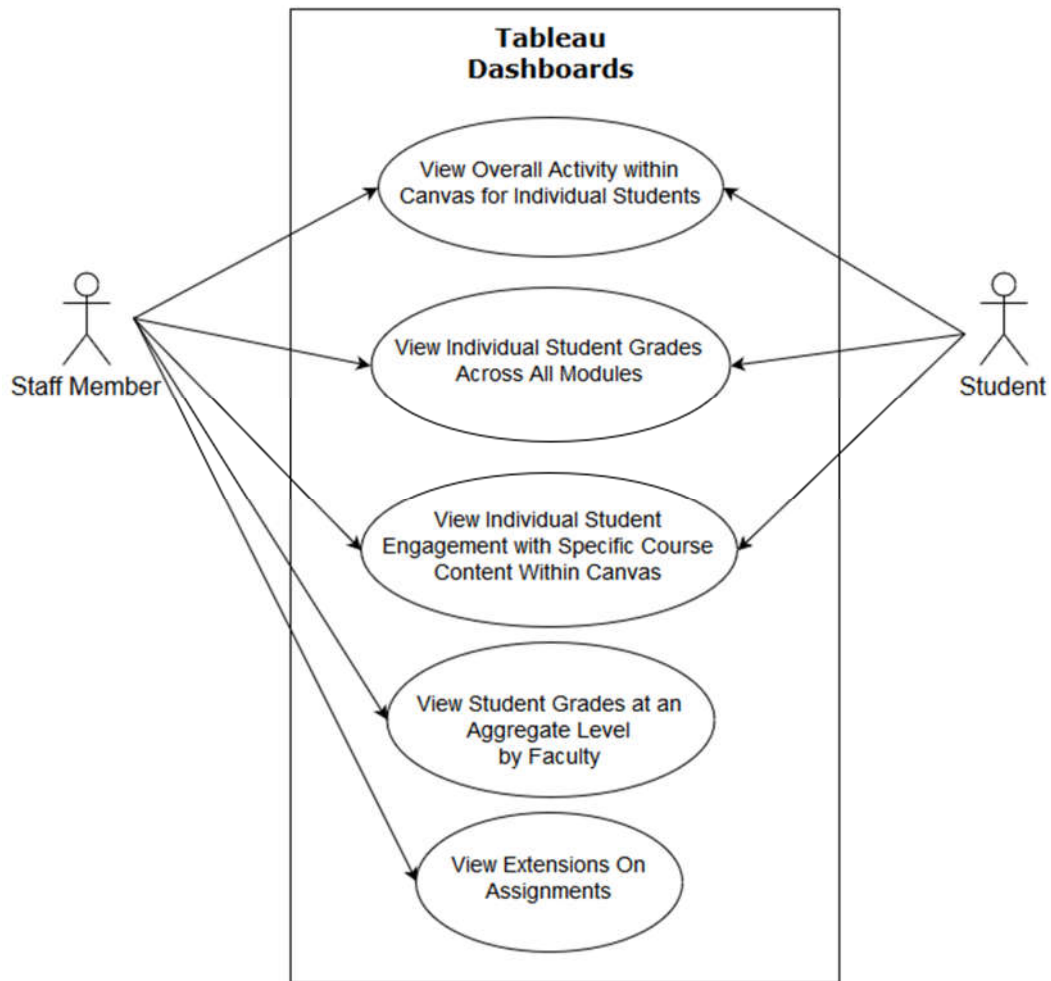


Figure 6. Use Case Diagram for the First Iteration.

3.2 Second Iteration

3.2.1 User Requirements

The requirements from the first iteration and the data available to meet those requirements were considered. The results of the first iteration were shown to the client and Canvas Analytics was further investigated for more reporting ideas. Discussions of the above then resulted in the below changes being made to the requirements.

Added:

- Create an aggregate-level report showing all user engagement with specific module content.
- Create a report showing individual student assignment statuses (late/on time).

Removed:

- Data about extensions in Canvas is not consistent enough to generate a report as multiple methods are used. In some cases, new assignments are generated for users who have been granted extensions and in others the original submission date is amended. This information cannot be aggregated to report on accurately.
- Student grades at an aggregate level by faculty is also too inconsistent due to different grading conventions. The necessary information is also incomplete as not all grades are uploaded to Canvas such as examinations and graded practical work. SITS data may need to be used to respond to this requirement.

3.2.2 Use Cases

Use cases were revised alongside the changes to requirements for the second iteration, detailed in the diagram below:

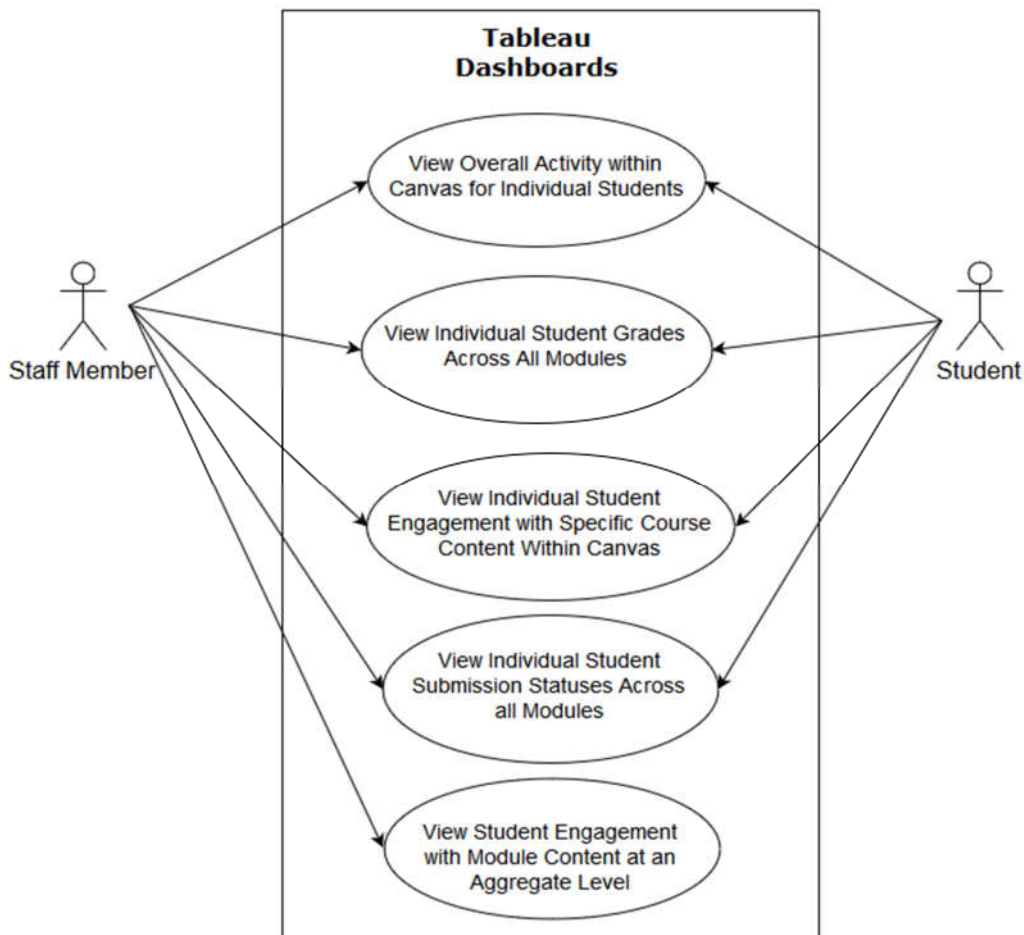


Figure 7. Use Case Diagram for the Second Iteration.

3.3 Third Iteration

3.3.1 User Requirements

The requirements from the second iteration were implemented and presented to the client. During this iteration some technical faults were highlighted and amended, and one more requirement was introduced:

- Create a report that details when an individual student has accessed assignment feedback across all modules. This is intended for staff use only.

3.3.2 Use Cases

The use case was then revised as per the diagram detailed in the figure below.

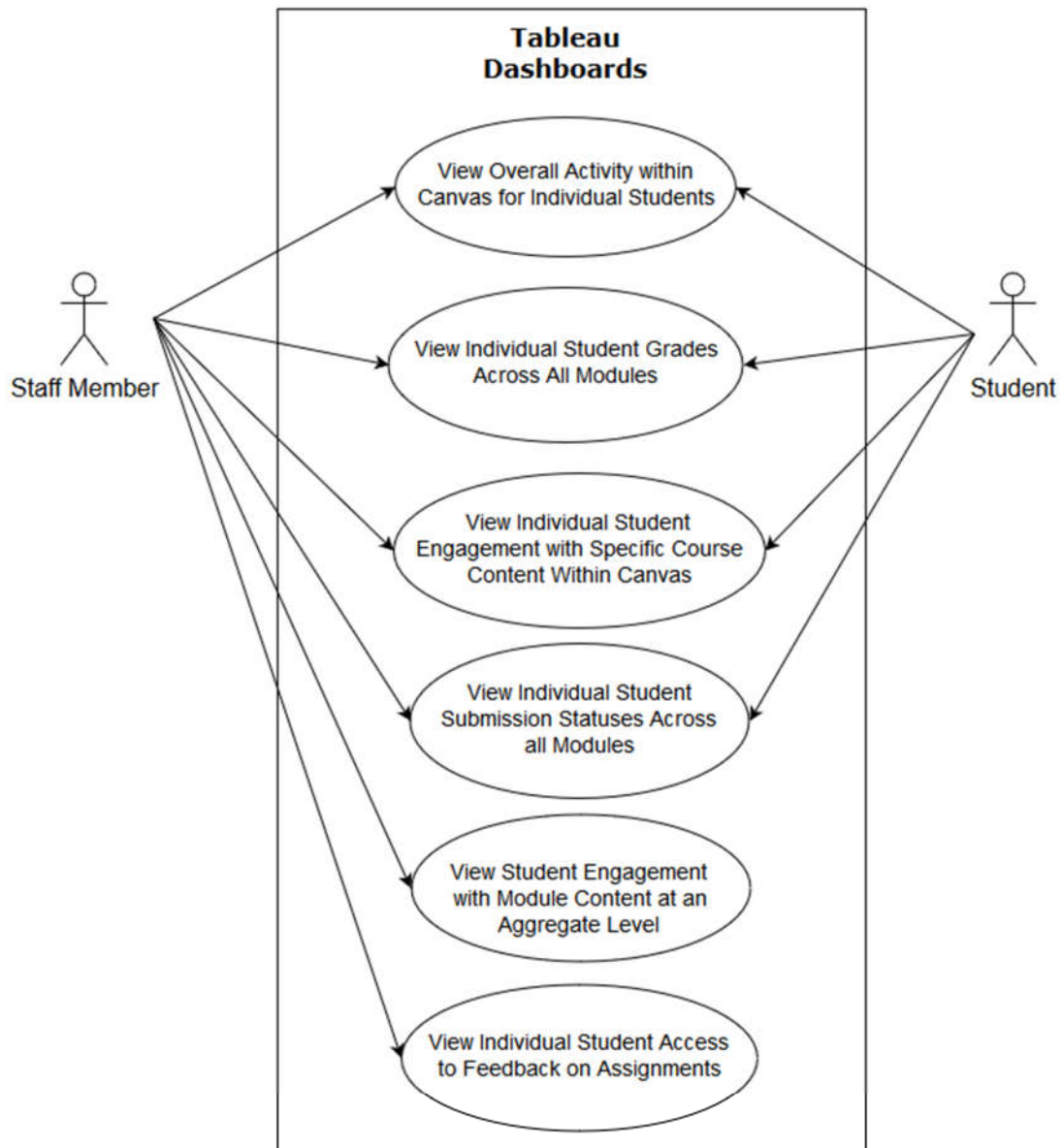


Figure 8. Use Case Diagram for the Third Iteration.

3.4 Non-functional Requirements

3.4.1 Dashboard Requirements

User requirements for the project were paramount for capturing the intended functionality of the dashboards, however, additional consideration was also given to appearance.

It was highlighted in discussion with the client that alternative reporting methods to those readily available to teaching staff are often too complex to interpret or difficult to obtain. Therefore, the following were identified as non-functional requirements of the dashboards:

- The dashboards must be visually appealing.
- The dashboards must be easy to use and navigate without excessive instruction.
- The information presented on the dashboards must be easy to interpret.

3.4.2 Database Design Requirements

Before reports and visualisations can be implemented in Tableau dashboards, Live Events data must first be re-structured for use in the project. To keep the database maintainable for future development, additional requirements are also listed for the structuring of Live Events data:

- No superfluous fields should be retained; keeping the number of fields in the new Live Events table to a minimum.
- The new Live Events table should be structured in such a way that it is easy to interpret for future development.

3.5 Requirement Analysis

The requirements as outlined in this chapter have implications for many elements of the project design: the data needed to meet the requirements will dictate how Live Events will be structured in the database; the queries used to return the necessary results will be dictated by the questions set out in the requirements, and; the overall design of the dashboards will be based on the needs of the users. These implications are set out in the two figures below; one which outlines the design implications for the appearance of the dashboards and one for the implications for the design of the data.

Requirement	Implication for Design of the Dashboard
Easy to use.	GUI features need to be implemented in the dashboard to enable users to easily interact with the data.
Visually appealing.	Content needs to be presented using a simple layout with good use of colour and styling for the visualisations.
Easy to interpret.	Titles, labels and instructions need to be written in language that is easy to understand.
Individual dashboards for staff and students.	Dashboards need to be altered as appropriate for each intended user group, including wording, content and access to data.
Staff members should see data about individual students.	Data needs to be searchable by 'student' in the staff dashboard(s) as appropriate.

Students should only see individual data about themselves.	Data must not be searchable by 'student' in the student dashboard(s).
Staff members should see aggregated module data about access to module content.	Data should be searchable by 'module' in the staff dashboard(s) as appropriate.
Users (staff/students) should see individual students' activity within Canvas by day similar to Canvas Analytics.	Number of Live Events by day should be presented in a graph on the dashboard(s) with dates along the bottom and height of bars dictated by amount of activity.
Accessible on the web or through Canvas on a desktop.	Dashboards need to be made in an appropriate size that can be embedded and viewed in a desktop browser.

Figure 9. Requirements and Implication for the Appearance of the Dashboards.

Requirement	Implication for the Design of the Data
Easy to interpret.	Names of fields need to be translatable into language that is easily recognisable by users.
Staff members should see data about individual students & students should see individual data about themselves.	Live Events must contain fields that can identify students or point to Canvas Data in such a way that students can be identified.
Staff members should see when students have accessed assignment feedback.	Live Events must contain fields that can identify assignments as well as time and date of access. They should be joined with Canvas Data appropriately to indicate the assignment and module it belongs to.
Users (staff/students) should see individual students' activity within Canvas by day.	Rows from the Live Events table will need to be counted and grouped by student and day. An identifier is needed to count distinct instances.
Staff members should see aggregated module data about overall access to module content.	Live Events must contain fields that can identify specific module content or point to Canvas Data in such a way that the module and the content can be identified. Live events will also need to be counted in this case to indicate number of 'clicks'/times accessed.
Users (staff/students) should see when an individual student has accessed specific module content.	Live Events must contain fields that identify the student, module content, date and time. These fields will need to be joined with Canvas Data appropriately to identify the associated module and grouped by student.

Figure 10. Requirements and Implications for the Structure of the Data.

3.6 Achievable Goals

Having taken into consideration the requirements for the project, a list of achievable goals was created as a framework for assessing the success of the final product:

1. Live Events should be restructured in such a way that the data can be queried in conjunction with Canvas Data using Tableau.
2. Data about individual students and aggregate module data should be made available to staff members, but students should only see individual data about themselves.
3. The dashboards should provide data about individual student grades.
4. The dashboards should provide data about individual student assignment submissions.
5. The dashboards should provide data about individual student activity in Canvas.
6. The dashboards should provide data about individual student access to module content.
7. The dashboards should provide aggregated data about all access to module content.
8. The dashboards should provide data about when students are accessing assignment feedback.

4 Data Structuring and Analysis

4.1 Canvas Data

4.1.1 Retrieval and Processing

Canvas Data is dumped into flat files ready for download from the Canvas Data Portal once every 24 hours. The files become available at 2am Mountain Time (MST) – that is 9am local time (BST) – and contain a complete refresh of all of the data.

To ensure data consistency, the relevant tables in the university's MySQL database are also emptied and updated with the new data set once every 24 hours. Updating such a large amount of data could cause disruptions to daily operations, so updates to the database are scheduled every evening at 10pm BST. It is important to be aware of this fact as the implications are that Canvas Data can be at least 36 hours old at any given time.

Each table in Canvas Data is downloaded as an individual tab-separated text file which is then parsed and uploaded to the database using a script containing SQL commands as below:

```
LOAD DATA LOCAL INFILE 'e:/canvas/data/quiz_question_answer_dim.txt'  
INTO TABLE quiz_question_answer_dim  
FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' IGNORE 1 LINES;
```

In the above example, the `quiz_question_answer_dim` table is populated using the data from the downloaded text file at location

`'e:/canvas/data/quiz_question_answer_dim.txt'`. The last line of the command determines that each new field is separated by a tab (`\t`), new rows are initiated with new line characters (`\n`), and one line in the text file is to be ignored. The first line of each text file contains header information that is already embedded in the table which is why it is omitted. The same SQL commands are used for each of the 88 tables, replacing the file location for the appropriate text file and table name for the appropriate table.

The above script for retrieving and processing the data was implemented by the development team at the university and is how the MySQL database for Canvas Data remains populated. Access to this database was granted for use in this project.

4.1.2 Data Model

Canvas Data implements Dimensional Modelling (DM) methodology developed by Ralph Kimball to create a star schema[17].

The star schema is a popular technique used to organise data warehouses as it offers fast and effective handling of simple queries. Data is structured in fact and dimension tables. Fact tables hold foreign keys to dimension tables and quantitative data about an event or occurrence, and dimension tables hold the descriptive (often text-based) attributes about that data. When creating a diagram, a single fact table is often placed at the centre with the referenced dimension tables connected to it creating the shape of a star. This is where the star schema gets its name.

The DM method stands in contrast to Entity-Relationship (ER) modelling used for relational databases. ER modelling uses normalisation to divide data into distinct tables containing only data about each individual object or concept. This method greatly reduces duplication of data as each table is self-contained but results in the existence of many tables containing little data. In order to query a relational database, foreign key links must be used to join the tables back together and queries can become very complex depending on how many tables the necessary

data is held in. The benefits of ER are of a transactional nature; rows in tables can be updated and deleted quickly whilst only affecting the relevant data. However, this separation makes it difficult to query in an intuitively. DM aims to resolve this problem of complex queries by providing flat, de-normalised dimension tables that contain all of the descriptive information needed in an intuitive way. Fact tables, which can then be used to join together different dimensions, are usually normalised in a similar fashion to relational databases and provide all foreign keys for each dimension.

Canvas Data consists of 39 fact tables and 49 dimension tables. Each are identifiable by the suffix “_fact” or “_dim” respectively. All attributes and foreign key relationships in the tables are described in the documentation on the Canvas Data Portal.

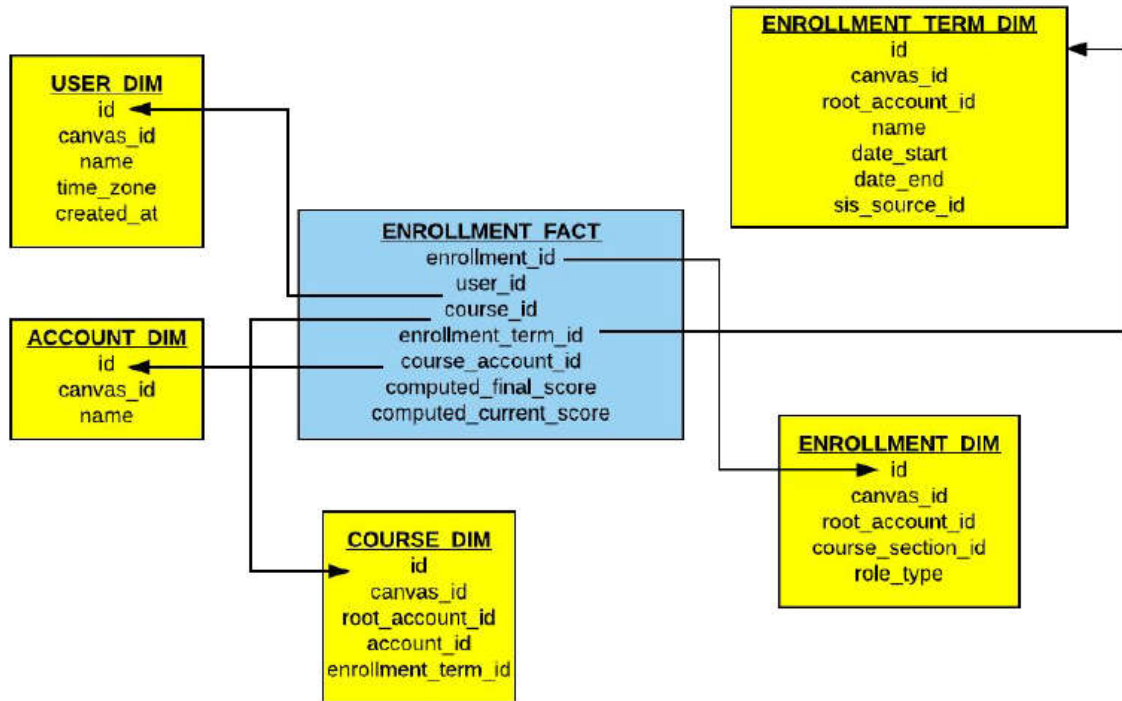


Figure 11. Diagram representing the star schema on the Enrollment_fact table.

4.1.3 Data Exploration

Before being able to run any sort of analysis on Canvas Data, it is important to identify the tables that will be the most useful in answering the questions set out by users.

While gathering user requirements, it became clear that the following areas were of particular interest: data about students; grades; assignments and assignment submissions; modules and module content; assignment feedback; and Canvas activity.

The documentation on the Canvas Data Portal was used to investigate the fields that may meet the above requirements. It was the intention that the student activity requirement would be met by Live Events so Canvas Data was not investigated for that purpose at this stage. Live Events are detailed in Section 4.2 of this chapter.

It is worth noting that there was no specific table or field found for assignment feedback, a problem which is tackled in chapter 5.

Figure 12 outlines data needed to meet requirements and where it can be found in Canvas Data.

Required data	Canvas Data Table	Fields of Interest and Descriptions.
Students	User_dim	<i>id</i> - Unique surrogate ID for the user. <i>global_canvas_id</i> – Canvas ID used to link this data to other Canvas Data tables and Live Events. <i>name</i> - Name of the user.
	Pseudonym_dim	<i>user_id</i> - Global Canvas ID of the user. <i>unique_name</i> - University network username. <i>SIS_user_id</i> - Unique identifier for the user set by the Student Information System (SIS), in this case, student number.
Student grades	Enrollment_fact	<i>user_id</i> - Canvas ID of user. <i>course_id</i> - Canvas ID of the module. <i>computed_current_score</i> - current module score. <i>computed_final_score</i> - Final score for the module
	Submission_fact	<i>submission_id</i> - Canvas ID of the submission. <i>assignment_id</i> - Canvas ID of the assignment. <i>course_id</i> - Canvas ID of the module. <i>user_id</i> - Canvas ID of the user. <i>published_score</i> - final score that was published for the assignment.
Modules	Course_dim	<i>id</i> - Canvas ID of the module. <i>name</i> - Title of the module. <i>code</i> - Module code. <i>SIS_source_id</i> - Module code provided through SIS.
Module content	File_dim	<i>id</i> - Canvas ID of the file. <i>display_name</i> - Name of the file (as appears on the uploaded document). <i>course_id</i> - Canvas ID of the course it belongs to. <i>content_type</i> - Type of file: PDF, PowerPoint, etc.
	Wiki_page_dim	<i>id</i> - Canvas ID of the wiki page. <i>title</i> - Name of the page. <i>body</i> - contents of the page.
Assignments	Assignment_dim	<i>id</i> - Canvas ID of the assignment. <i>course_id</i> - Canvas ID of the module. <i>title</i> - Title of the assignment. <i>due_at</i> - The date that the assignment is due. <i>points_possible</i> - Top score possible for the assignment. <i>workflow_state</i> - Status of the assignment, e.g. Published or unpublished.
Assignment submissions	Submission_dim	<i>id</i> - Canvas ID of the assignment submission. <i>assignment_id</i> - Canvas ID of the assignment. <i>course_id</i> - Canvas ID of the module it belongs to. <i>submitted_at</i> - Date and time of submission. <i>graded_at</i> - Date and time the submission was graded. <i>workflow_state</i> - Current state of the submission, e.g. submitted, unsubmitted, graded, pending_review.

Figure 12. User Requirements Matched with Canvas Data.

Once the necessary fields in Canvas Data were identified for use in the project, SQL queries were constructed to test if the fields could meet the reporting needs of the users. Two cases are outlined below:

Grades

One requirement identified by users was to see student grades across all modules to gain an overall view of how individual students are performing.

The *enrollment_fact* table was first investigated for use as it holds calculated current and final scores for each module. However, upon inspection of the data, the two fields are null throughout the dataset. This could be due to the way that Canvas is used at the university, as calculated module scores is a function that can be turned off.

The *submission_fact* table was used to meet this requirement instead, as it provides the published score for each assignment. The below SQL query was used to aggregate data about all assignments and their scores:

```
SELECT user_dim.`name` AS `Student Name`,
course_dim.`name` AS `Module Name`,
assignment_dim.title AS `Assignment Title`,
submission_fact.published_score AS `Mark`,
assignment_dim.points_possible AS `Out Of`
FROM canvas.assignment_dim
JOIN canvas.submission_fact
JOIN canvas.user_dim
JOIN canvas.course_dim
WHERE assignment_dim.id = submission_fact.assignment_id
AND user_dim.id = submission_fact.user_id
AND course_dim.id = assignment_dim.course_id
ORDER BY `Student Name`;
```

The logic of the above query is to select students, the modules they are enrolled in, titles of assignments given to them, the marks they received and the total possible score for each assignment by joining the *assignment_dim*, *submission_fact*, *user_dim* and *course_dim* tables by their respective IDs. It is ordered by student name to view each student's assignments together.

The query returned the following results:

Student Name	Module Name	Assignment Title	Mark	Out Of
[Names Redacted]	ITNP070 - Introduction to Netwo...	Practical 4	8	8
	ITNP070 - Introduction to Netwo...	Practical 3	8	13
	ITNP070 - Introduction to Netwo...	Practical 1	14	15
	ITNP023 - Foundations of Inform...	Coursework: A presentati...	27	30
	ITNP033 - Database Principles an...	Coursework [Assianment 1]	89	100
	ITNP033 - Database Principles an...	Examination [Assianment 2]	NULL	100
	ITNP023 - Foundations of Inform...	Examination [Assianment 2]	NULL	100
	ITNP070 - Introduction to Netwo...	Examination [Assianment 4]	NULL	100
	ITNP070 - Introduction to Netwo...	Class Test [Assianment 2]	NULL	100

Figure 13. Results of SQL query selecting student grades.

The results in the above figure indicate one student's assignments and grades across three modules, indicating that the query was successful in retrieving the data needed to meet this requirement.

Assignment Submission Statuses

Another requirement was to identify whether students are submitting their assignments late or on time. *Assignment_dim* contains information about when assignments are due and *submission_fact* contains information about when they were submitted, so the below SQL query was used to explore this possibility of answering this question:

```
SELECT user_dim.`name` AS `Student Name`,
course_dim.`name` AS `Module Name`,
assignment_dim.title AS `Assignment Title`,
assignment_dim.due_at,
submission_dim.submitted_at,
submission_dim.workflow_state
FROM canvas.assignment_dim
JOIN canvas.submission_dim
JOIN canvas.user_dim
JOIN canvas.course_dim
WHERE assignment_dim.id = submission_dim.assignment_id
AND user_dim.id = submission_dim.user_id
AND course_dim.id = assignment_dim.course_id
AND submission_dim.submitted_at < assignment_dim.due_at;
```

The logic of the above SQL command is to select the student name, module, assignment title, assignment due date, submission date, and workflow state from the *assignment_dim*, *submission_dim*, *user_dim* and *course_dim* tables joined on their respective IDs.

The query selects only for assignments that were submitted prior to the due date as declared in the clause “*submission_dim.submitted_at < assignment_dim.due_at*”. The aim of creating a query such as this is to test if it is possible to create a selection that will return only details about assignments submitted on time.

The following results were returned:

Student Name	Module Name	Assignment Title	due_at	submitted_at	workflow_state
[Names Redacted]	LAWU9RE - Research Methods (...)	Dial Essav - Video submissi...	2017-12-01 12:00:00	2017-11-30 23:25:46	graded
	LAWU9RE - Research Methods (...)	Westlaw Advanced Certifi...	2017-12-01 12:00:00	2017-09-28 17:43:27	graded
	LAWU9RE - Research Methods (...)	Westlaw Basic Certificate	2017-12-01 12:00:00	2017-09-28 17:55:05	graded
	SPSU9SB - Managing the Busines...	Assianment 1 - Career Pro...	2018-02-06 12:00:00	2018-02-05 23:15:42	graded
	SPSU9RP - Research Project in S...	Self assessment form	2018-03-13 12:00:00	2018-03-12 11:10:31	submitted
	SPSU9RP - Research Project in S...	Final Research Project	2018-04-10 11:01:00	2018-04-10 08:59:38	submitted
	SPSU9R7 - Readinas in Soorts St...	Dopina Testina in Sport	2017-10-31 12:00:00	2017-10-31 00:43:33	graded
	SPSU9RP - Research Project in S...	Research Project Propoosal	2017-11-07 12:01:00	2017-11-03 03:50:14	graded
	SPSU9SP - Sports Policv (2017/8)	Assianment 1: Writina a P...	2017-10-11 11:01:00	2017-10-11 02:56:56	graded
	CRMU9TZ - Dissertation (2017/8)	Dissertation [Assianment 1]	2018-04-17 14:00:00	2018-04-13 13:16:25	submitted

Figure 14. Results of the SQL query selecting for data about assignments submitted on time.

As can be seen from the figure above, the query was successful in returning the intended results. In each case, the due date is later than the submission date and the state of the assignment is consistent. The query can then also be easily amended to return data about late assignments by changing the comparison operator from less than (<) to greater than (>).

To ensure that the results of the queries reflected what actually happened within Canvas, a few rows were selected for to test for consistency. Administrator privileges in Canvas were used to inspect grades and submissions in Canvas and compared to the data in the results.

The reason to do such testing is two-fold:

First, to ensure that the joins on the tables were correct and represent reality. Canvas IDs have similar formats across all dim and fact tables and share the same numbers, which means that incorrect joins could easily be made if the wrong fields are compared. Doing such a test ensures that the correct assignment was paired with the correct module, and that the student was truly enrolled on that module and submitted the assignment.

Secondly, to ensure that the submission dates and grades in Canvas reflect those reported in the results of the queries. This information is what users are most interested in for reporting purposes, so it is important that it is accurate.

The data selected for testing matched Canvas in both cases, indicating that the queries were formulated correctly. These queries were used as the basis for retrieving data in Tableau as outlined in Chapter 5.

4.2 Live Events Data

The second part of this chapter discusses Live Events. One of the goals set for the project was to re-structure the way that Live Events are stored in order to make the data more easily usable in conjunction with Canvas Data. For this reason, the chapter will outline how the original Live Events were retrieved and the method used to process them into a new format before discussing the data model based on the newly restructured data.

4.2.1 Retrieval

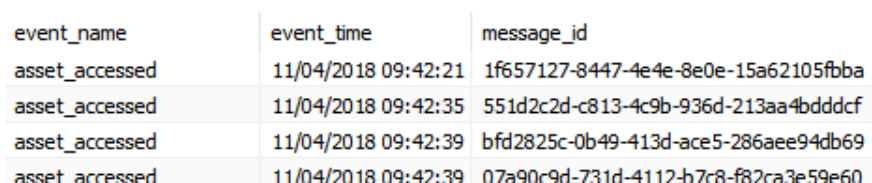
Live Events are sent to the university using Amazon SQS. Unlike Canvas Data which is updated once every 24 hours, Live Events are provided in a best-effort fashion as soon as the event takes place and added to the university's message queue.

Each message in the queue contains a body and attributes about the event that occurred. The attributes supplied are a unique message ID, the event time and the event name. The body of the message contains all of the necessary information about the event in IMS Caliper 1.1 format which makes use of JSON-LD (JavaScript Object Notation for Linked Data).

The messages are requested from the queue through an API call every couple of minutes, uploaded to the MySQL database, and deleted from the queue. This was set up by the university prior to the project.

Live Events are stored in one table with in the MySQL database containing the same fields as the original message: *Message_id*, *Event_time*, *Event_name* and *Body*.

Due to the *body* field containing JSON data, Tableau was unable to read this field and only returned the *event_name*, *event_time* and *message_id* fields when querying the table:



event_name	event_time	message_id
asset_accessed	11/04/2018 09:42:21	1f657127-8447-4e4e-8e0e-15a62105fbba
asset_accessed	11/04/2018 09:42:35	551d2c2d-c813-4c9b-936d-213aa4bdddcf
asset_accessed	11/04/2018 09:42:39	bfd2825c-0b49-413d-ace5-286aee94db69
asset_accessed	11/04/2018 09:42:39	07a90c9d-731d-4112-b7c8-f82ca3e59e60

Figure 15. Screenshot from Tableau Displaying the Original Live Events Table.

In order to view the *body* field and investigate the data, the SQL command below was used in MySQL Workbench:

```
SELECT JSON_PRETTY (Body) FROM canvas.live_events;
```

The above command produces a readable version of the data by formatting it with new lines and indents as appropriate. Below is an example of the body of one event that was retrieved in this way:

```
{
  "data": [
    {
      "id": "urn:uuid:2532f2f7-b145-40f6-a711-9d3d9c6f2f47",
      "type": "NavigationEvent",
      "actor": {
        "id": "urn:instructure:canvas:user:110220000000013203",
        "type": "Person",
        "extensions": {
          "com.instructure.canvas": {
            "entity_id": "110220000000013203",
            "user_login": "[REDACTED]",
            "root_account_id": "110220000000000001",
            "root_account_lti_guid": "1vLzdxkTi0cmLkTFe10E8qq7p08I0cg9fYtef00:canvas-lms"
          }
        }
      },
      "edApp": {
        "id": "http://canvas.stir.ac.uk/",
        "type": "SoftwareApplication"
      },
      "group": {
        "id": "urn:instructure:canvas:course:110220000000002675",
        "type": "CourseOffering",
        "extensions": {
          "com.instructure.canvas": {
            "entity_id": "110220000000002675",
            "context_type": "Course"
          }
        }
      },
      "action": "NavigatedTo",
      "object": {
        "id": "urn:instructure:canvas:course:110220000000002675",
        "name": "grades",
        "type": "Entity",
        "extensions": {
          "com.instructure.canvas": {
            "entity_id": "110220000000002675",
            "asset_type": "course",
            "asset_subtype": "grades"
          }
        }
      },
      "session": {
        "id": "urn:instructure:canvas:session:05cce84e66bab4dbc252bd4263e6d6ae",
        "type": "Session"
      },
      "@context": "http://purl.imsglobal.org/ctx/caliper/v1p1",
      "eventTime": "2018-04-11T09:42:21.000Z",
      "extensions": {
        "com.instructure.canvas": {
          "version": "1.0.0",
          "hostname": "canvas.stir.ac.uk",
          "request_id": "e80b88e4-a631-4c05-b842-84721b84bd02",
          "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36,"
        }
      },
      "membership": {
        "id": "urn:instructure:canvas:course:110220000000002675:Learner:110220000000013203",
        "type": "Membership",
        "roles": [
          "Learner"
        ],
        "member": {
          "id": "urn:instructure:canvas:user:110220000000013203",
          "type": "Person"
        }
      },
      "organization": {
        "id": "urn:instructure:canvas:course:110220000000002675",
        "type": "CourseOffering"
      }
    }
  ],
  "sensor": "http://canvas.stir.ac.uk/",
  "sendTime": "2018-04-11T03:30:15.851Z",
  "dataVersion": "http://purl.imsglobal.org/ctx/caliper/v1p1"
}
```

Figure 16. A sample of the Live Events *Body* field for an “Asset_accessed” event.

Figure 16 allows us to better see the structure of the data. JSON data is commonly structured as a list of attribute-value pairs separated by commas, held between two curly brackets {}. In this case the majority of the data are held in an array called “data” which is indicated with square brackets []. Each attribute that has multiple attribute-value pairs to describe it contains these pairs nested within another set of curly brackets. An example of this can be seen in the figure below, showing nested attribute-value pairs about the “actor” attribute in an event. Further nesting can also be seen for the “extensions” attribute held within the “actor” attribute:

```

"actor": {
  "id": "urn:instructure:canvas:user:110220000000013203",
  "type": "Person",
  "extensions": {
    "com.instructure.canvas": {
      "entity_id": "110220000000013203",
      "user_login": "[REDACTED]",
      "root_account_id": "110220000000000001",
      "root_account_lti_guid": "1vLzdxkT10cmLKTFeE10E8qq7p0BI0cg9fYtef00:canvas-1ms"
    }
  }
}

```

Figure 17. An example of nested JSON data describing the “actor” in an event.

The *body* field in Live Events is rarely consistent. Due to the NoSQL nature of JSON, attributes that are relevant to one type of event but not relevant to another are omitted or added as appropriate. The result is between 30 and 50+ attribute-value pairs for any given *body* field in Live Events. However, basic structure for JSON data held in the *body* field can be predicted to some extent by looking at the *event_name* field. This field provides information about the type of event that occurred and dictates the structure of the *body* to some extent.

There are 27 different possible values for *event_name*, containing the likes of “asset_accessed”, “logged_in”, “logged_out”, “course_created” and “attachment_updated”.

Understanding the structure of the data is important in order to identify how it can be used in conjunction with Canvas Data. By inspecting the JSON data it can be seen that Canvas ID values are given for some attributes, in the example above a user ID is given of “110220000000013203” which can be a reference to the *user_dim* table in Canvas Data.

The task at hand was then to process this data in such a way that it could be used in Tableau to join Live Events with Canvas Data.

4.2.2 Data Selection and Processing

A decision was made to convert the JSON data into a new MySQL table for Live Events. There were two main reasons for choosing to format the data this way: first, it was a definite way of ensuring that Tableau was able to interface with both datasets, and second, to make the dataset consistent by structuring it into a determined set of fields containing details about each event. The question was then to select the data that would go into this new MySQL table.

4.2.3 Selecting an Event Type

The first aspect considered for data selection was the type of event that would be collected; this would help dictate a structure for the table.

When referring to the requirements set out in Chapter 3, the reporting needs of the users focused on 1) how much student activity there was within Canvas and 2) access to specific module content. An event type was needed that would satisfy both requirements. During discussions with the client, all values occurring in the *event_name* field were considered. It was established that the chosen value should represent the bulk of the events that occur in Canvas, as to ensure that the majority of the events are captured. In order to investigate this, the events were counted and grouped by distinct *event_name* to produce Figure 18:

asset_accessed	10009367
logged_in	479275
24nrolment_state_updated	230721
attachment_created	175690
submission_updated	113065
grade_change	96747
24nrolment_created	61317
enrollment_state_created	61317
group_membership_created	35942
user_account_association_created	31113
wiki_page_created	21981
logged_out	21970
submission_created	19505
assignment_updated	14878
enrollment_updated	14288
wiki_page_updated	14287
quiz_submitted	8131
discussion_entry_created	7583
discussion_topic_created	6908
attachment_deleted	6117
assignment_created	5515
group_created	3170
course_created	971
wiki_page_deleted	685
group_category_created	614
attachment_updated	116
syllabus_updated	47

Figure 18. The count of distinct records per *event_name* ordered from most to least.

The figure above illustrates that the “asset_accessed” value occurs the most times with over ten million records, making it a good candidate for selection. However, further consideration was given to the other values before a decision was made.

When inspecting the numbers, it can be seen that the “logged_in” event occurs the second most times; in discussion it was decided that “logged_in” does not denote a meaningful event within Canvas as any user could log in but not interact with the LMS in any useful way. This type of event only indicates one action that the user made at the start of their session. This value was removed from consideration.

The next few values on the list indicate actions that are predominantly performed by teaching or administrative staff instead of students. These are values such as “enrollment_state_updated”, “attachment_created”, “grade_change” and “enrollment_created”.

These values were deemed not fit for purpose as student interaction with Canvas was of interest, not administrative events.

There were some values that appeared low on the list that were helpful to consider, such as “discussion_entry_created” which only occurred 7583 times. This type of event details when a student or staff member has posted in a discussion board. Although this type of activity is meaningful, it was decided that there was not enough data to base the structure of the new table around this type of event as the attribute-value pairs are specific to discussion post data. This could possibly change in future if there is a shift in how Canvas is used, as discussion boards are only used in a handful of modules at present. The aim of this selection is to choose an event that gives the most consistent representation of activity across all modules and this event type does not fit this requirement.

Ultimately, the “asset_accessed” event was selected for use because it contained the most records (thus detailing the most activity) and provided information about users clicking on specific objects in Canvas. The referenced objects include areas in discussion boards and other module content, so even though some activity would be lost (such as the posting of a discussion board entry), the activity surrounding it would still be captured. Therefore, the “asset_accessed” event met both requirements of indicating how much activity was taking place and what module content was being clicked on.

4.2.4 Selecting the Fields

The second aspect of data selection was to choose the fields for the new Live Events table based on the structure of “asset_accessed” type events. One of the requirements set out for the project was to ensure that the number of fields was kept down to a minimum to eliminate redundancy, so it was important to select the fields that were consistent across each event and provided the most useful information.

A group of 20 “asset_accessed” events were extracted and compared for their contents. A comprehensive list of the most consistent attribute-value pairs was created which identified 39 possible fields for selection. The documentation for Live Events and the IMS Caliper 1.1 format were then used to identify what the values in these pairs represent. The full list can be found in Appendix 1.

Due to the way that the data is structured in JSON, the fields named in this section indicate the path of where the attribute is nested in the data. For example, the ID of the actor is named in this document as *actor_extensions_entity_id* because the *entity_id* attribute is nested within the *extensions* attribute which is in turn nested in the *actor* attribute.

There were a number of attributes identified that remained identical for each event. These attributes were the likes of *action* (“NavigatedTo”), *type* (“NavigationEvent”), *session_type* (“Session”) and *sensor* (“<http://canvas.stir.ac.uk/>”). These fields were excluded from consideration as they provided no useful information.

A nested group of attributes were found in the attribute *membership*, detailing the object and actor of an event taking place within the group context menu in Canvas. These attributes only duplicated the details of the object and user and did not provide meaningful information.

Additionally, there were attributes identified that provided information about the browser used to access Canvas and the user’s IP address; although this information could be useful it is not of interest for this project and was excluded from consideration.

The user requirements were also consulted for assistance in choosing the necessary fields as per Figure 10 in Chapter 3. The most important details needed to meet the requirements were those pertaining to the student that initiated the event, the object they clicked on, and the time and date of the event.

Taking into consideration all of the above factors, the fields below were chosen for the new Live Events table:

Attribute	Description
<i>data_id</i>	Unique identifier for this event.
<i>actor_extensions_entity_id</i>	The unique Canvas ID of the actor creating the event.
<i>actor_extensions_user_login</i>	The university log in for the user (university network username).
<i>object_name</i>	The name of the object that was accessed during this event in Canvas, e.g. an assignment, attachment, course, grades, etc.
<i>object_extensions_entity_id</i>	The unique Canvas ID of the object that was accessed. The specific Canvas ID that this field points to is determined by the object name and asset type fields, e.g., if <i>object_name</i> and is "course" then this field points to the ID in the Canvas Data course_dim table.
<i>object_extensions_asset_type</i>	This field provides the higher-level navigational context information of the object being accessed. E.g., if the event was triggered by a user clicking on the "home" page within a course navigation menu, object_name would contain "home" and object_extensions_asset_type would contain "Course".
<i>session_id</i>	The session identifier for this event. Can be used to correlate events in the same session for a user.
<i>eventTime</i>	Time and date that the event took place.

Figure 19. Selected Live Events attributes and their descriptions.

There was one field that was not selected as an attribute to keep in the new Live Events table but was still important for the process of creating it, namely: *actor_type*. This attribute described the actor initiating the event, holding one of two values: "Person" or "SoftwareApplication". If the value of the attribute was "SoftwareApplication", it indicated that the event occurring was an API call from Instructure or SITS.

Since the project focused on human interaction with Canvas, the events where a software application created the event were excluded from the set. The method used for handling this type of filtering and the creation of the new table is discussed in section 4.2.5.

4.2.5 Creating and Populating the New Live Events Table

With the necessary fields selected and the structure of the JSON data identified, a new table for Live Events was created: Live_Events2.

The table was created using the same naming conventions as outlined in Figure 19 to maintain consistency with the JSON data and assigned VARCHAR datatypes of appropriate size. The reason for using VARCHAR datatypes was to maintain the consistency between the format of Live_Events2 and that of Canvas Data, allowing for easy comparisons of the fields.

Once the table was created, a script was written using PHP to retrieve and reupload data from Live Events into Live_Events2.

The PHP script had to process the data in a particular way, best expressed in this series of steps below:

1. Retrieve a row of Live Events data where the *event_name* is "asset_accessed".
2. Assign the *Body* field containing the JSON data to a new PHP variable.
3. Use PHP function `json_decode()` to convert the JSON into PHP-readable format.
4. Find the *actor_type* attribute in the JSON data. If the value is not "Person", skip to the last step.
5. Assign the selected fields to new PHP variables.
6. Insert the values of the variables into the Live_Events2 table using an SQL command.
7. Loop through this process again for the next row until there are no more rows returned.

An important part of the process was identifying the individual attributes within the PHP object containing all of the JSON data. The PHP statements used to pick out these attributes are detailed in the table below:

Live_Events2	PHP
data_id	<code>\$obj->data[0]->id;</code>
actor_extensions_entity_id	<code>\$obj->data[0]->actor->extensions->{'com.instructure.canvas'}->entity_id;</code>
actor_extensions_user_login	<code>\$obj->data[0]->actor->extensions->{'com.instructure.canvas'}->user_login;</code>
object_name	<code>\$obj->data[0]->object->name;</code>
object_extensions_entity_id	<code>\$obj->data[0]->object->extensions->{'com.instructure.canvas'}->entity_id;</code>
object_extensions_asset_type	<code>\$obj->data[0]->object->extensions->{'com.instructure.canvas'}->asset_type;</code>
session_id	<code>\$obj->data[0]->session->id;</code>
eventTime	<code>\$obj->data[0]->eventTime;</code>

Figure 20. Location of JSON attributes stored in the PHP variable \$obj.

Another crucial element to this process was ensuring that null values were handled appropriately. In some cases, the *object_name* value would be empty, in which case the Live_Events2 field was assigned the same value as *object_extensions_asset_type*. This was done because the latter field provides the same type of information but at a higher level in a navigational context. In so doing, the *object_name* field would always be populated with information about the object being clicked on which is useful if the field needed to be used in isolation.

4.2.6 Redesigned Live Events Data Model

The original format of Live Events made it near impossible to create a diagram sufficient to explain the relationships between the attributes of Live Events and Canvas Data. Live_Events2 was created with a more straightforward relationship to Canvas Data; acting similarly to a fact table in a star schema by containing only qualitative data or foreign key references to Canvas Data tables. However, there was one dependency that complicated the structure of these relationships, namely: the dependency that the *object_extensions_entity_id* field has on the fields *object_name* and *object_extensions_asset_type*.

As described in Figure 19, the Canvas ID value provided in the *object_extensions_entity_id* field identifies the object that was clicked on within Canvas. The problem is that this ID points to a different Canvas Data table depending on what type of object was clicked on. In order to be able to ascertain which table to join on Live_Events2, one must look at the name of the object and asset type for an indication.

The diagram below depicts this relationship between Live_Events2 and Canvas Data:

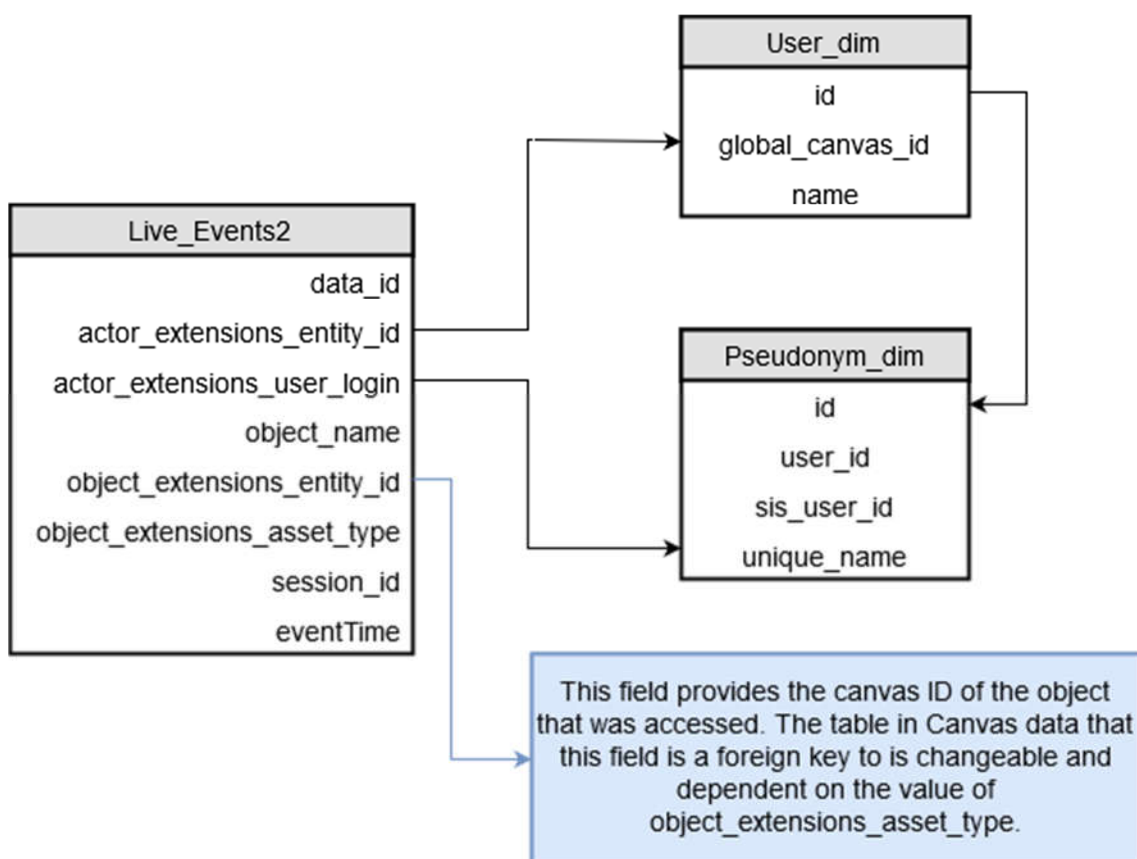


Figure 21. Diagram representing the Relationships between Live_Events2 and Canvas Data

The *object_extensions_entity_id* field was then capable of providing an answer to the requirement of detailing access to specific module content, however, this complex relationship between it and the other fields needed to be understood first. Upon further inspection, it was found that the *object_extensions_asset_type* field provided the best indicator to the table being referenced. *Object_name* was not suitable for this as it provided information that was too low-level and not represented in Canvas Data (i.e. there are no tables describing “home” pages, but there is a *course_dim* table and courses each have home pages in Canvas).

The *object_extensions_asset_type* field provided 15 distinct values; 12 of which were direct indications to a Canvas Data table. These relationships are identified in the diagram in Figure 22.

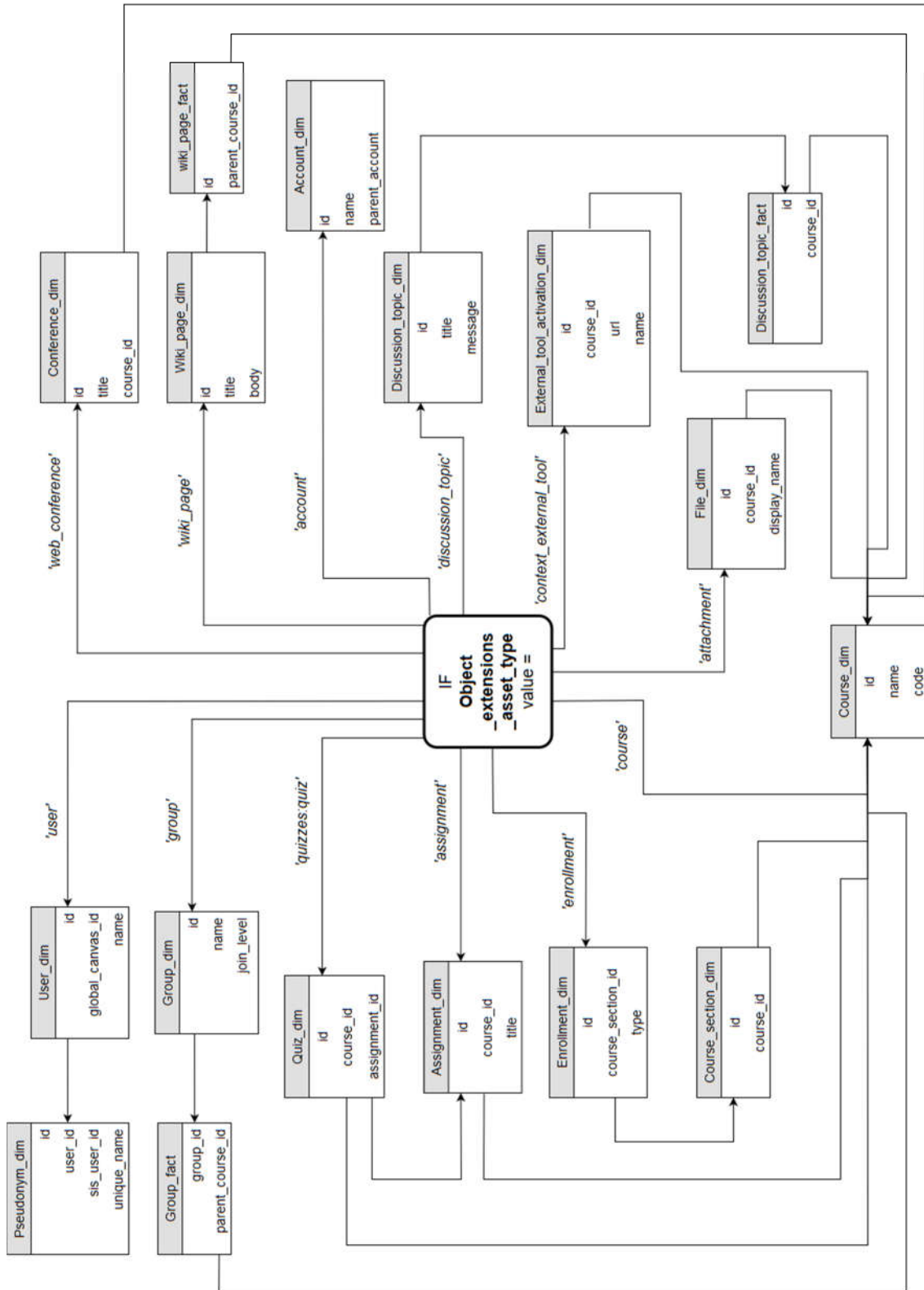


Figure 22. Relationships between *object_extensions_entity_id* and Canvas Data.

The three fields that did not point to Canvas Data tables were identified to be references to external tools or content tags that are not hosted in Canvas and have been omitted above.

4.2.7 Data Exploration

Once the relationships between Live_Events2 and Canvas Data were identified for use in the project, SQL queries were constructed to test if joining the two datasets could meet the reporting requirements of users. Two such cases are outlined below:

Activity by Day

A reporting requirement that was met previously in part by Canvas Analytics was giving an indication of user activity by day, however, this was provided by module instead of indicating overall activity.

In order to answer this requirement with Live_Events2, the total activity by day can be counted and grouped by user. The SQL used to accomplish this is below:

```
SELECT user_dim.`name`,
SUBSTRING(live_events2.eventTime, 1, 10) as `Date`,
COUNT(live_events2.data_id) as `Number of Events`
FROM canvas.user_dim
JOIN canvas.live_events2
WHERE live_events2.actor_extensions_entity_id = user_dim.global_canvas_id
GROUP BY `Name`, `Date`;
```

The above query first creates a substring of the *EventTime* field, curtailing it so that only the date of the event remained. This then allowed for the dates to be grouped and counted by user. The *data_id* field was used for counting as this provides unique instances of each event.

The query returned the results below:

Name	Date	Number of Events
[Names Redacted]	2018-04-15	14
	2018-04-15	3
	2018-04-14	6
	2018-04-11	4
	2018-04-12	9
	2018-04-11	9
	2018-04-12	19
	2018-04-14	6
	2018-04-12	21

Figure 23. Results of SQL query selecting for number of events by day.

As indicated in the figure above, the query was successful in returning the intended results, as each user’s activity was counted and listed per day.

Access to Module Content

The requirement to identify when users are interacting with specific module content in the form of teaching materials (PowerPoint presentations, PDFs, and other uploaded files) was investigated using the relationships identified in Figure 22.

It was identified that when the value of *object_extensions_asset_type* is “attachment”, the *object_extensions_entity_id* points to the *file_dim* table. By matching the Canvas IDs in these two

tables, it was possible to retrieve the exact file name that users had clicked on in each event. This reporting capability was not previously provided by Canvas Analytics.

The SQL query below was used to test this relationship:

```
SELECT user_dim.`name`,
file_dim.display_name,
live_events2.eventTime
FROM canvas.live_events2
JOIN canvas.file_dim
JOIN canvas.user_dim
WHERE live_events2.object_extensions_entity_id = file_dim.id
AND live_events2.actor_extensions_entity_id = user_dim.global_canvas_id
AND live_events2.object_name = 'attachment';
```

The following results were returned:

name	display_name	eventTime
[Names Redacted]	Chapter 3.docx	2018-04-12T14:20:12.000Z
	Report.docx	2018-04-13T00:59:43.000Z
	SHSU13A Multimorbidity Case Study Frank FI...	2018-04-13T16:46:44.000Z
	Report.docx	2018-04-13T07:45:45.000Z
	Theme 1 Study Guide 2018.docx	2018-04-13T13:11:52.000Z
	HISUM06 module handbook spring 2018.pdf	2018-04-14T15:46:31.000Z
	D1Tut.pdf	2018-04-12T12:04:17.000Z
	PSYU912-2612612.doc	2018-04-12T13:27:41.000Z
	6 - DM Project.pdf	2018-04-12T13:30:27.000Z
	Cameron McGhee - Dissertation.pdf	2018-04-12T08:59:15.000Z
	Monitoring and Evaluation.pdf	2018-04-15T11:13:31.000Z

Figure 24. Results of SQL query selecting for students, file names and the times they were clicked on by the student.

The above results indicate a successful use of the query, by returning specific file titles and the times and dates they were clicked on by each student. The above could also be expanded to add the module title by joining on the *course_dim* table, as the *file_dim* table provides a field for *course_id*.

As with the testing that took place for queries on Canvas Data in the previous section, these two queries were also checked for consistency with Canvas and in each case matched the circumstances in reality. Although Canvas does not provide exact same details about the items that were clicked on by students, it is possible to check if this behaviour is consistent by checking if the student is in fact enrolled on the specific module and if they have access to the material. Canvas does provide some access information by providing the URLs students have accessed, and if this is inspected it is consistent with the results of the queries.

The exploration of Canvas Data and Live_Events2 shows that the two datasets are able to be joined and can meet requirements of users. The following chapter explores how this information was used to create visualisations in Tableau to make this data accessible to users.

5 Visualising with Tableau

Tableau is a powerful reporting and visualisation tool that can be used to create similar queries to those explored in Chapter 4 using a GUI instead of writing queries manually. The understanding gained in the previous chapter formed a basis for constructing queries and retrieving the necessary data for visualisation in Tableau. The process undertaken to create the final dashboards is outlined in the sections below.

5.1 Initial Investigation

Prior to the commencement of the project, the reporting capability of Tableau was tested by the client who had created a simple workbook using similar parameters to those investigated in section 4.1.3 of this dissertation. The workbook contained a dashboard displaying student grades.

The work done by the client was useful in order to form a basis on which the current project could be built, allowing the client to show an example of what the expected outcomes from this project should be and how Tableau can be used. The workbook was provided for this project to use in an initial investigation into Tableau.

During this investigation, a few important features of Tableau were highlighted. Although there is an option to write custom SQL, the GUI provides an interface to join Canvas Data tables as in the image below. It can be seen that joins are represented as circular links between tables:



Figure 25. Canvas Data tables joined in Tableau.

It is also possible to have different types of joins in Tableau, each with their own depiction. Join clauses are also set using this interface by selecting attributes from a drop-down list:

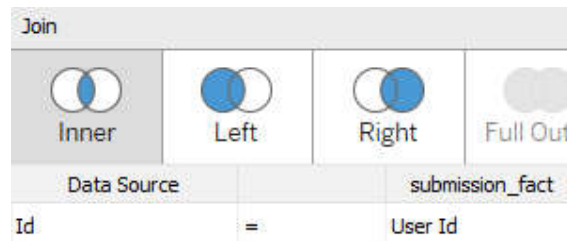


Figure 26. Joins and join clauses depicted in Tableau.

Fields can be renamed and split from the data source tab for use in the workbook:

The screenshot shows a Tableau data source table with columns: 'Course Id (Assign...', 'Title', 'Description', and 'Due At'. The table contains several rows of data. A context menu is open over the first row, showing options like 'Rename', 'Reset Name', 'Copy Values', 'Hide', 'Aliases...', 'Create Calculated Field...', 'Create Group...', 'Split', 'Custom Split...', and 'Describe...'. The 'Rename' option is highlighted.

Course Id (Assign...	Title	Description	Due At
110220000000000592	essay 1		29/06/2017 15:00:00
110220000000000592	Presentation		30/06/2017 22:59:00
110220000000000592	Workshop Dis		17/08/2017 22:59:59
110220000000000592	test essay		19/06/2017 14:00:00
110220000000000592	Unnamed quiz		null
11022000000001431	Assignments i		null
11022000000001431	Coursework [A		11/12/2017 17:01:00
11022000000001431	Assignments i		null

Figure 27. Field options Menu displayed in Tableau.

Calculated fields can also be implemented, whereby a new field is created using a calculation against another field (or fields). Comparisons between fields can be made by using comparison operators. The calculations are often formed by using “If” statements terminated by the word “END”, such as in the example below:

```

Calculated Field

IF [Course Id]=[Canvas Id (Course Dim)] THEN "Correct" END

```

Figure 28. Example of a Calculated Field.

Worksheets are created by dragging and dropping fields into rows and columns. Although the example below is of a basic text-based report, there are options to create graphs, charts and other visualisations by making changes in the “Marks” panel on the left-hand side of the worksheet. Above marks is the “Filters” panel where certain values can be included, excluded, or applied as a usable filter in the worksheet.

Name	Module	Title	Mark	Out Of
Victoria Szymanska	ITNP001 - Principles and Practice of Programming (2017/8)	Class Test	97	100
		Crossword Assignment	80	100
		Practical checkpoints	100	100
	ITNP023 - Foundations of In..	Coursework: A presentation on topics in UX [As..	27	30
	ITNP033 - Database Principl..	Coursework [Assignment 1]	89	100
	ITNP070 - Introduction to Networking and Webscripting (2017/8)	Coursework [Assignment 1]	95	100
		Practical 1	14	15
		Practical 3	8	13
		Practical 4	8	8
	ITNPBD6 - Data Analytics (2..	Coursework [Assignment 1]	77	100

Figure 29. Example workbook displaying student grades.

The above worksheet can then be added to a dashboard, where filters can be used as interactive elements. In the example below, a student’s name can be entered and selected from a drop-down list to view an individual students’ grades.

Enter Student Name

Victoria Szymanska

Student Grade Review

Name	Module	Title	Mark	Out Of
Victoria Szymanska	ITNP001 - Principles and Practice of Programming (2017/8)	Class Test	97	100
		Crossword Assignment	80	100
		Practical checkpoints	100	100
	ITNP023 - Foundations of In..	Coursework: A presentation on to..	27	30
	ITNP033 - Database Principl..	Coursework [Assignment 1]	89	100
	ITNP070 - Introduction to Networking and Webscripting (2017/8)	Coursework [Assignment 1]	95	100
		Practical 1	14	15
		Practical 3	8	13
		Practical 4	8	8
	ITNPBD6 - Data Analytics (2..	Coursework [Assignment 1]	77	100

Figure 30. Student Grade Review Example Dashboard.

5.2 Dashboards

Once the initial investigation into Tableau was conducted, the project focused on the creation of new workbooks and dashboards. The final product resulted in 4 distinct dashboards, which are each described in their own sections below.

5.2.1 Assignments and Grades Dashboard

The purpose of this dashboard was to meet two requirements, namely: displaying student grades from across all modules and displaying assignment submissions across all modules. The initial worksheet created by the client was expanded on in the final version of this dashboard.

In chapter 4 the SQL queries needed to answer these specific needs were tested, and these SQL queries formed the basis on which this dashboard was built. The changes to these queries and the process followed to create this dashboard are outlined below:

After the investigation phase, an initial dashboard was created during the first iteration of this project. It contained two simple reports that were both filtered by student name. During the second iteration of the project, it was decided that it would be better to select students by their student ID number instead of by their names. There were two reasons for this decision: to ensure their anonymity and to avoid having the data of users with identical names being brought together.

In order to be able to sort by the student ID of any user, the *pseudonym_dim* was needed as it contains the field with the student ID values. The below joins were made using Tableau:



Figure 31. Joins used for the Assignments and Grades Dashboard

The *SIS_user_id* field of the *pseudonym_dim* table containing student IDs was renamed appropriately in Tableau and then applied as a filter for the data.

A feature in Canvas Analytics that was deemed useful was the colour-coding of assignment statuses “late”, “on time” and “not submitted”. In order to try and emulate this concept, a calculated field was created that compared the due date and submission date for assignments in this report:

```
IF [Due At]-[Submitted At] <0 THEN "Late"  
ELSEIF [Due At]-[Submitted At] >0 THEN "On Time"  
ELSEIF ISNULL([Submitted At]) THEN "Not Applicable"  
ELSEIF ISNULL([Due At]) THEN "Not Applicable"  
END
```

Figure 32. Calculated field used for Assignment Submission Statuses.

The values in this calculated field were then coded with colours and added as a column at the end of the report. Each line now represented the data about when an assignment was due, when it was handed in, and the associated colour for if it was late, on time, or not applicable.

During the second iteration of the project it was also identified that the assignments listed in the submission status report were not all meant to be submitted by the user, and in some cases, should not be available to the user in the first place. Upon investigation, it was noticed that the *workflow_state* field of the *assignment_dim* table contained the details about whether or not the assignment was published. A filter was created to only view published assignments.

This new assignment status and grades dashboard was then reviewed in user testing, and changes were made on the back of the feedback that was received.

The first item that was amended in line with feedback was the amount of text in the submission date and due date fields. Each field originally contained the date and time down to second precision, which caused the text to look cluttered and unreadable. These fields were shortened in the workbook.

Another amendment made in line with user feedback was the addition of a visualisation for student grades. The original student grade report contained only textual information about the score received and the top score possible, but due to a lack of consistency in grading schemes, it would be hard to tell if the score was good or not just by glimpsing at the grades.

To resolve this issue that was identified, a new calculated field was created to calculate the percentage of the score received on any given assignment, which was then used to create a bar graph depicting this percentage alongside the scores. This gave a more easily recognisable visual representation of what the student's performance looks like.

General feedback from users was that the text in the dashboard was not very legible, and that the harshness of the white background was making it difficult to read. It was pointed out that dyslexic users struggle with text against a harsh white background, so in order to make the dashboard more accessible and easier to view for all users, a pale blue background and dark blue text were implemented.

In the end, this was the final dashboard that was created:

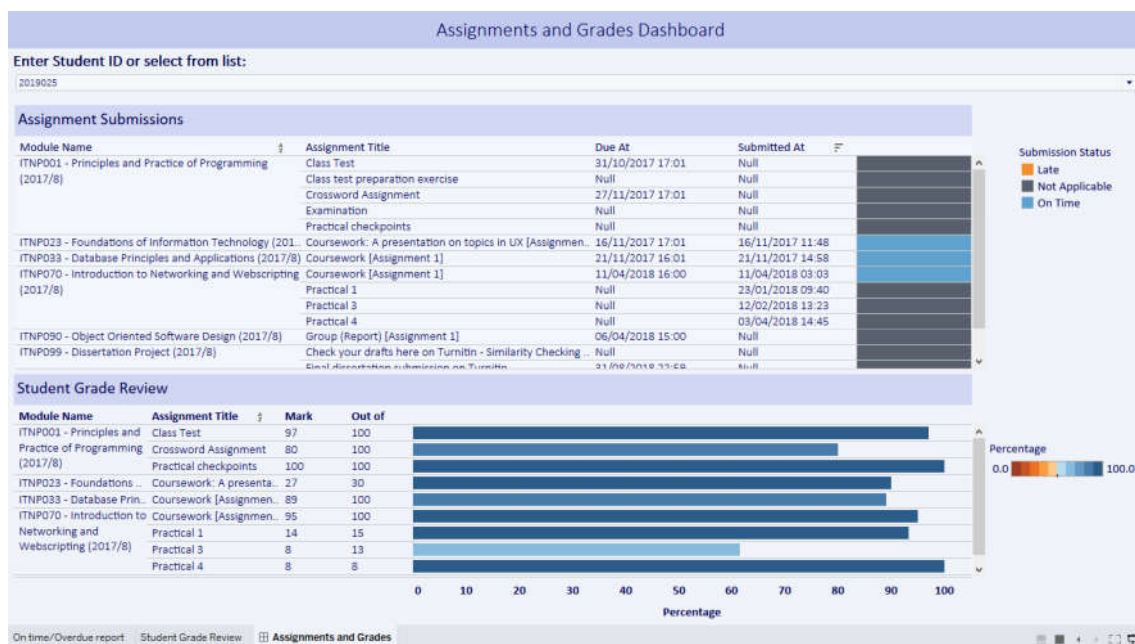


Figure 33. Assignments and Grades Dashboard.

5.2.2 Access and Activity Dashboard

The access and activity dashboard aimed to meet two more requirements set out by users: the ability to view user activity by day, and their access to specific course content.

Once again, the SQL queries in the previous chapter were used as a basis for this dashboard. Similar alterations were made as discussed for the Assignments and Grades dashboard to allow for filtering by student ID instead of by student name.

The resulting joins are depicted below:



Figure 34. Joins used for the Access and Activity Dashboard.

In order to be able to group events by day, the *EventTime* field was split into two fields: date and time. The date field was then formatted as a continuous “date” type within Tableau so that it could be used create a graph similar to that in Canvas Analytics. By making this data type conversion, a graph was able to be plotted with dates along the bottom and aggregated events (the number of clicks) along the side, indicating the number of clicks on any given day in the dataset in a bar graph.

The same general colour scheme and layout of the Assignments and Grades dashboard was also applied to this one, ensuring consistency throughout the dashboards.

The original versions of this dashboard contained more complicated wording that was not as easily understood by users, so that feedback was used to change the wording of the likes of “Number of Events” to “Number of Clicks” and to give reports more descriptive titles.

A report was embedded into the dashboard displaying results very similar to that of the SQL query that was used to select users, the event time and the file names that were accessed. A column was added providing the module title. This was done to provide the contextual information of the module the file belongs to, which could aid in quicker identifying what the file is. One drawback of receiving the exact name of the file as titled in the upload is that it does not always make sense, so hopefully adding in this context will be useful.

The final dashboard looks like this:

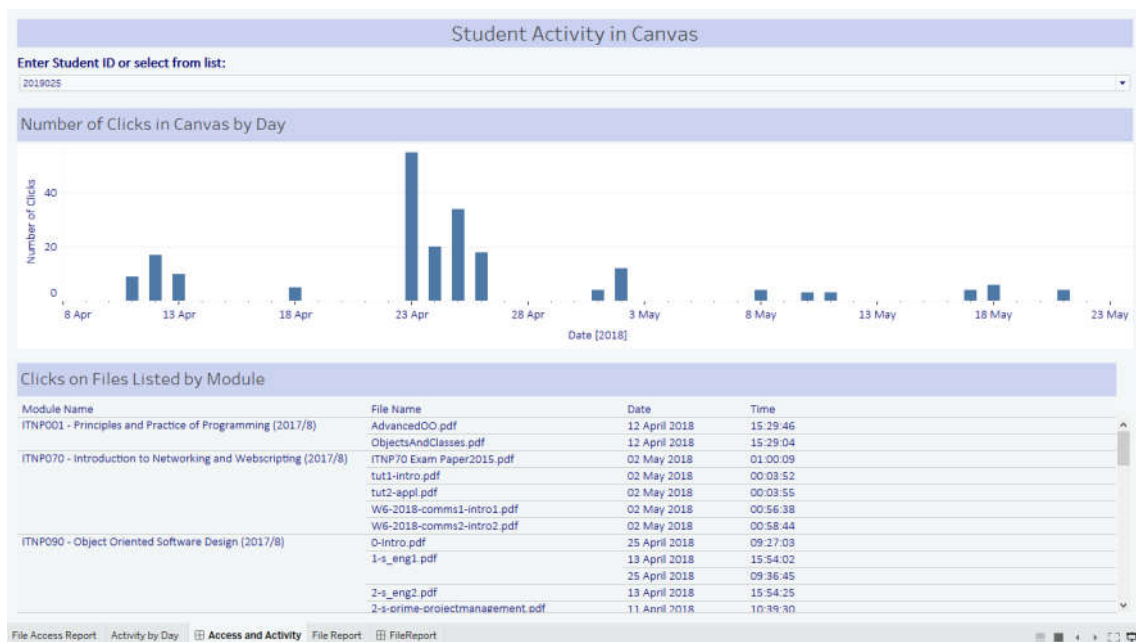


Figure 35. Access and Activity Dashboard.

5.2.3 Frequency of Clicks on Module Content Dashboard

This specific dashboard was created with the intention of being used by staff members only, to meet the requirement of seeing how often their module content is accessed aggregated by all students.

Because the query is so similar to that of the file access report in the previous section, the same joins from the same workbook were used:



Figure 36. Joins used for the Frequency of Clicks on Module Content Dashboard.

The query was rather simple, as live events had to be counted and grouped by file and module

Two versions of the report in this dashboard were created during the initial iteration of the project; one showing a simple table listing the names of files and how many times they were clicked on, and another showing a graph where each file is represented by a block and the colour and size of the block is dictated by the number of clicks on each file. The latter was preferred as an option and was used in the final dashboard instead.

The original chart used the colour green on a spectrum from dark to very light green. During user testing this was found to be difficult to read and possibly troublesome for the colour blind. The colour scheme was changed to go from oranges to blues, where orange colours represent less clicks, and blue colours represent more clicks.

The module name field was applied as a filter so that it could be used for selection and placed at the top of the screen to remain consistent with the other dashboards.

The colour scheme of the dashboard was also amended to match the others with a pale blue background.

The final dashboard looks like this:

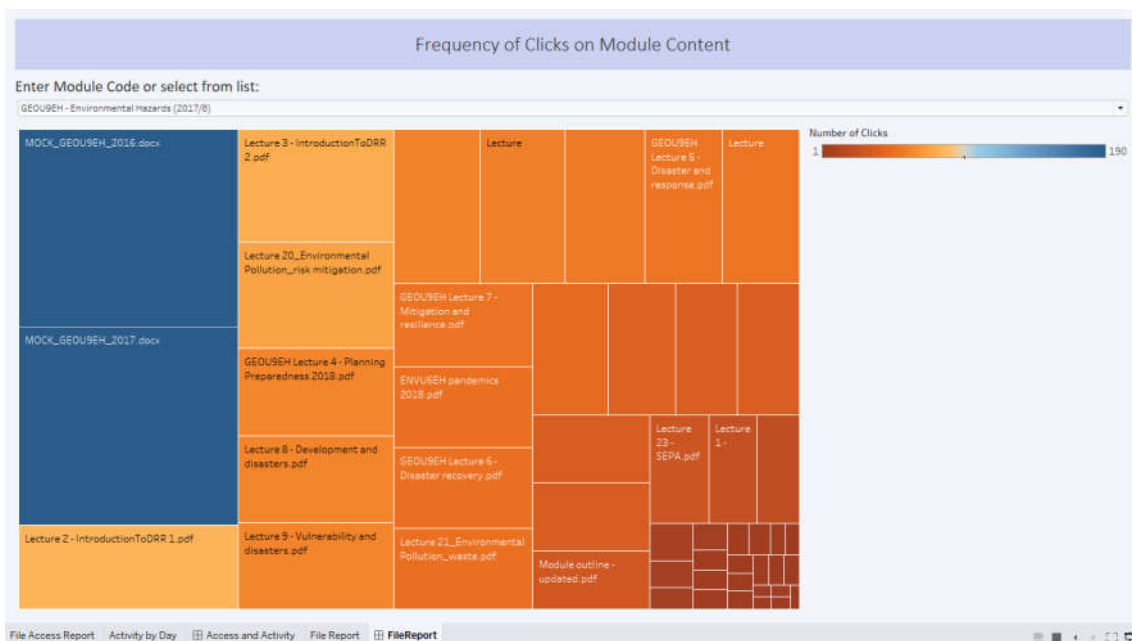


Figure 37. Frequency of Clicks on Module Content.

5.2.4 Feedback Access Dashboard.

A user requirement that was introduced in a later iteration of the project was a report that could indicate when a student was accessing feedback given on their assignments.

This was a more difficult query to answer as upon inspection of the Canvas Data, the feedback given by instructors was not stored as an identifiable object in the data. When using Canvas, the feedback provided for an assignment is usually displayed in the assignment area and grades are displayed in the grade area.

Discussions were had with the client about what could be used as a substantial proxy measure, and it was decided that a report that details when a student has accessed an assignment after it has been marked would be an appropriate approximate measure.

In order to construct this query, details about the date the assignment was graded are needed, as well as the module it belongs to, the date and time it was accessed, and the user who submitted the assignment. The following joins were made to answer this query:



Figure 38. Joins used for the Feedback Access Dashboard.

It was important to ensure that only the instances when a user had accessed the assignment after the graded date were displayed, so a calculated field was created and used as a filter to exclude the data that didn't fit this pattern.

During user testing, participants commented on how it would be useful to see the due date listed in the report, as this would give an indication into how long a time period passes between submission and grading of assignments. This was added.

The consistent colour scheme of pale blues was also applied to this dashboard.

The final result looks like this:

Student Feedback Access Dashboard					
Enter Student ID or select from list:					
2517008					
Student Clicks on Graded Assignments					
Module Title	Assignment Title	Due At	Graded At	Accessed At	
EDUU9E3 - Pedagogy I: The Nature of Learning and Teaching (2017/8)	Coursework [Assignment 1]	08/12/2017 15:01	12/12/2017 12:03	18/04/2018 15:17	
	Essay [Assignment 2]	27/11/2017 12:01	12/12/2017 15:14	18/04/2018 15:17	
FREU9A4 - Contemporary French Language and Cultures (2017/8)	Class Test [Assignment 2]	30/03/2018 16:00	17/04/2018 10:52	20/04/2018 16:35	
				20/04/2018 16:36	
	Essay [Assignment 5]	13/04/2018 16:00	15/04/2018 11:35	21/04/2018 20:35	
SPAU9A4 - Advanced Spanish and Latin America..				07/05/2018 10:28	
	Language Examination [Assignme..	Null	09/05/2018 14:58	07/05/2018 10:29	
				14/05/2018 07:03	

Figure 39. Feedback Access Dashboard.

6 Evaluation

6.1 Evaluation Against Set Goals

In section 3.6 of this dissertation, achievable goals were set out in a list to be used for evaluation of the final product. This list took into consideration the user requirements for the project and will now be revisited to compare to the final result.

Goals:

- Live Events should be restructured in such a way that the data can be queried in conjunction with Canvas Data using Tableau.
- Data about individual students and aggregate module data should be made available to staff members, but students should only see individual data about themselves.
- The dashboards should provide data about individual student grades.
- The dashboards should provide data about individual student assignment submissions.
- The dashboards should provide data about individual student activity in Canvas.
- The dashboards should provide data about student access to module content.
- The dashboards should provide aggregated data about all access to module content.
- The dashboards should provide data about when students are accessing feedback.

Evaluation:

The project was successful in restructuring the JSON data from Live Events into a MySQL table. Live_Events2 fits well into the Canvas Data model, acting as a fact table which holds references to other Canvas Data tables. Most importantly, Tableau is capable of reading and joining both sets of data together.

Individual dashboards were created and tailored for each type of user. Two of the dashboards were intended for both user types, but the option to select a student ID was removed from the student dashboards to ensure they could only see their own data.

The Assignments and Grades Dashboard was created successfully showing all grades across all modules for individual students. A calculated percentage for each grade was also represented as a bar graph beside the scores to give a better visual indication about their performance.

The Assignments and Grades Dashboard shows submission statuses for assignments completed by individual students, colour coded for “late”, “on time” and “not applicable”.

The Access and Activity Dashboard provides a graph displaying student activity by day. The height of the bars on the graph is dependent on the number of clicks in a day.

The Activity and Access Dashboard provides a report detailing when individual students have clicked on specific files across all modules.

The Frequency of Access to Module Content Dashboard provides a useful visualisation detailing the frequency of clicks on specific files uploaded to the module.

The Feedback Access Dashboard provides a report detailing when students are accessing their assignments after they have been marked. This was agreed as a substantial proxy measure for when students are accessing feedback.

Over all, the final product meets the goals set in Chapter 3 and meets the requirements set by the client. Further evaluation is carried out by looking at the dashboard and data more closely.

6.2 Evaluation of Dashboards

In order to fully evaluate the design of the dashboards, it is important to consider the factors that influence user experience. The eight golden rules of interface design[18] are used as a basis for evaluation the dashboards.

The rules:

1. Strive for consistency.
2. Seek universal usability.
3. Offer informative feedback.
4. Design dialogues to yield closure.
5. Prevent Errors.
6. Permit easy reversal of actions.
7. Keep users in control.
8. Reduce short-term memory load.

Evaluation:

The first rule of interface design is adhered to throughout. From a visual design standpoint, all of the dashboards use the same colour scheme, fonts and layout. The use of colour to convey meaning is particularly consistent across all of the dashboards, as oranges and reds are used to convey negative elements (low grades, late submissions, fewer clicks on module content) whereas blues are used to convey positive elements (high scores, submissions that are on time, more clicks on module content). The actions required to interact with each of the dashboards are consistent: a user or module needs to be selected by choosing a value from a drop-down list at the top of the screen. The language used throughout the dashboards is consistent, with fields labelled in a way that is easy to understand.

The dashboard design also adheres to the second rule of being universally usable. All dashboards are made easy-to-use for any user by providing only one sequence of actions that could be performed. This means that beginners can easily interact with the dashboard without needing to learn a confusing series of steps, and experienced users can quickly and easily make use of them too. An added feature enabled by Tableau itself is the use of keyboard shortcuts (for actions such as undo) which more experienced users can benefit from.

Rules 3 and 4 about offering informative feedback and designing dialogues to lead closure are in part met by the Tableau software itself. When a longer query is being executed, a window will appear on screen that prevents action from being taken until the query is complete. The user is required to wait for the box to disappear at which point the results of the query will show. Unfortunately, it is not possible to edit the dialogue boxes that appear on screen; such a function would be useful for providing easier to understand text for the user.

The design of the dashboard itself also creates a sensible dialogue, as the user is asked to “enter or select a student ID/module” and upon the completion of the task, the corresponding data is presented. The sequence of actions has a sensible start, middle and end.

Prevention of errors as per rule 5 is yet again adhered to by offering users a limited choice of actions. In this way, users cannot make non-sensible queries that could break the dashboards as the options presented to them will yield sensible results. By setting the filters in such a way

that a value needs to be selected from a list instead of entered freely, users cannot enter a student number or module code that does not exist, and if they do the list to select from would appear empty.

Tableau also permits the easy reversal of actions as per rule 6; every query can be easily undone by the use of the “undo” button or keyboard shortcut. The only issue with this is whether or not users are aware that this is an option as it is not made obvious. It is fairly simple for the user to manually undo their own actions by going back and selecting their intended value, however, it can be time consuming for queries that take a long time to execute.

The second-last rule about keeping users in control is where the dashboards may not be as effective as they could be. The dashboards are designed in such a way to respond to user input, however, because the permitted actions are so limited it may leave users feeling like they do not have enough control. The speed of some queries can also be frustrating, leaving the users waiting for the dashboard to respond. An improvement to the design of the data and the queries would improve the speed at which results are returned, so this is not necessarily a fault in the design of the dashboard, however it has an impact how it is perceived.

The last rule seeks to reduce short-term memory load in users. The design is successful in adhering to this rule. All of the necessary information pertaining to each dashboard is displayed in one screen, with headings and column titles remaining constantly visible if there is a scrollable list on screen. Legends for interpreting colour codes are constantly present on the dashboards so that users can easily find the information if necessary. The selected student or module title remains populated in the filter at the top of the dashboard, giving an indication as to which data is currently being viewed.

In conclusion, the overall design of the dashboards fares well when evaluated against the eight golden rules. There are improvements that could be made, but some limitations are imposed by the software itself which cannot be changed. A better underlying data structure would help improve the usefulness of the dashboards, but their appearance and design are effective and consistent.

6.3 Evaluation of the Data Design

The design of the restructured Live Events table was successful in meeting the crucial requirement of becoming usable in conjunction with Canvas Data. However, as discussed in the evaluation of the dashboards, some of the queries (in particular those making use of Live Events) would take longer to return results. This can be expected with databases that contain millions of records, however, it may be that the solution is not fit for this purpose if the project was scaled to a larger dataset as is the intention. The evaluation below discusses the positive and negative elements of the structure of Live_Events2.

First, the main positive: the newly structured table is simple and contains all necessary data to discern what happened during the click/event in Canvas. Redundant data has been eliminated and only events performed by human users – not software applications – is being stored. In removing the redundant data and by limiting the fields, Live_Events2 has become a table that specifically describes the things that users want to know about click data, namely: “who, what, and when”. Not only does this simplify the design of the table, it also makes it more intuitive to human users. The details provided by the object name and asset type fields describe the event in a useful way without needing to join it with Canvas Data; for example, it is still possible to tell how many times assignments were accessed, or when a user has clicked on announcements or grades in general without contextual information.

There is a negative element to this, however, as a loss of data occurs. The data in the Live_events2 table may be able to meet requirements, but there are 26 other types of events

that are not captured. All of the event types together may give a better picture of all user activity in Canvas, but as it stands, these different types of events could not be held in the same table if a MySQL format is used. A NoSQL JSON-LD solution was implemented for Live Events for this very reason in the first place; it allows different types of events to contain different sets of attribute-value pairs, without having a formal structure to adhere to.

Another negative in the design of the table is the naming conventions used. Although it is useful to know where the fields stemmed from in the JSON data, it is not entirely readable when inspected in isolation. The benefits of being able to give the fields aliases using SQL or Tableau are great, but it would be worthwhile permanently updating the names in the database to ensure that future use of the database is simple and self-explanatory.

A benefit of using a MySQL database is the opportunity to use tools provided by the database engine to make easier use of the data. For example, this project made use of indexes on certain fields to speed up queries. This method was implemented in small part due to time restraints, so a better understanding of indexes and how they may be used could improve the database in future.

It must be recognised that the current solution was chosen in part due to the resources made available by the university and how data was already being stored prior to the start of the project. When taking into consideration all of these factors, the selected solution may not be ideal, but the results met requirements in the quickest way possible with the resources available.

7 Conclusions

The project outlined in this dissertation was ultimately successful in achieving the main purpose of this undertaking: to give users access to Canvas Data and Live Events in a way that is understandable to them.

The dashboards that were created met all user requirements and most importantly surpassed Canvas Analytics by providing a more holistic view of the student experience. This was achieved by aggregating data from across all modules, but also by narrowing down specific instances of interaction with module content. This two-fold approach gave insight into both the overall activity and performance of a student but also specific instances of engagement.

It would be beneficial for the dashboards to be embedded in web pages for staff and students to use, however, this would require a big investment by the university to buy into Tableau Server. The dashboards are currently only viewable with a fully licensed Tableau Desktop product of which only two are currently held by the university for the purposes of this project.

The process of restructuring Live Events has not only allowed the project to meet requirements but has also given great insight into how the data is structured and how it relates to Canvas Data. This information is useful for future development, even if the results of this project are not used. Live Events are still developing as Canvas evolves, and more information should become available in future, including better documentation and more case studies about the institutions making use of it. The opportunities to explore this field are endless; this project focused on a subsection of the data which has opportunity to be expanded into a bigger picture.

One important issue that was highlighted during the process of this project was the inconsistencies in usage of Canvas among faculties. Canvas is still relatively new to the university (having only been introduced in the last year) so there are certain procedures that have not been formalised or implemented in Canvas. There is also the issue that some faculties use Canvas in a more light-touch manner than others. If a more consistent approach is used in future, the data will be able to provide even more insights into student performance and activity than at present.

Tableau is a powerful tool that can work with many different data types, so the problem encountered during this project was a unique one. There is a chance that if the JSON data had been stored in a NoSQL database, there would have been a way to query it in conjunction with the MySQL database using a different approach in Tableau. This may be worthwhile investigating for future development, as a NoSQL solution to the JSON data may open up possibilities that were unfortunately dismissed in this project.

The work completed as part of this project was focused mainly on descriptive analytics instead of predictive analytics, however, as the data set gets built up over time there is an opportunity to build an algorithm that could be used for predictive purposes in future. This dataset could also be used in conjunction with SITS and library data to give a better insight into student engagement as Canvas only tells part of the story.

The final result of this project is a set of dashboard prototypes that could one day be implemented for use and a wealth of understanding into Live Events. The University of Stirling being one of the first institutions to make use of this new dataset is an exciting look into the future and can only be an indication of greater things to come.

References

- [1] The Guardian. The Cambridge Analytica files: The Story so Far. (2018-03-26) Author: Patrick Greenfield. Available: <https://www.theguardian.com/news/2018/mar/26/the-cambridge-analytica-files-the-story-so-far> Accessed: August 2018.
- [2] Medium. How Does Spotify Know You So Well? (2017-10-10) Author: Sophia Ciocca. Available: <https://medium.com/s/story/spotify-s-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe> Accessed: August 2018.
- [3] Business Insider. Alexa's won Christmas this Year. (2017-12-26) Author: Steve Kovach. Available: <http://uk.businessinsider.com/amazon-alexa-top-ios-android-app-christmas-day-echo-sales-2017-12?r=US&IR=T> Accessed: August 2018.
- [4] Learning Analytics. Learning and Academic Analytics.(2011-08-05) Author: Siemens, G. Available: <http://www.learninganalytics.net/uncategorized/learning-and-academic-analytics/> Accessed: August 2018.
- [5] Quality Assurance Agency. Our Work in Scotland. Available: <http://www.qaa.ac.uk/about-us/where-we-work/our-work-in-scotland> Accessed: August 2018.
- [6] Enhancement Themes. Enhancement Themes Home Page. Available: <http://www.enhancementthemes.ac.uk/> Accessed: August 2018.
- [7] Enhancement Themes. About Page. Available: <http://www.enhancementthemes.ac.uk/about-enhancement-themes> Accessed: August 2018.
- [8] Enhancement Themes. Current Enhancement Themes. Available: <http://www.enhancementthemes.ac.uk/current-enhancement-theme> Accessed: August 2018.
- [9] Enhancement Themes. Collaborative Clusters. Available: <http://www.enhancementthemes.ac.uk/en/current-enhancement-theme/collaborative-clusters> Accessed: August 2018.
- [10] Jisc. Effective Learning Analytics Project Page. Available: <https://www.jisc.ac.uk/rd/projects/effective-learning-analytics> Accessed: August 2018.
- [11] Canvas Community. Document: Canvas Production Release Notes (2018-02-17). Available: <https://community.canvaslms.com/docs/DOC-14130-canvas-production-release-notes-2018-02-17> Accessed: August 2018
- [12] Canvas Community. Document: What is Canvas Data Live Events? (2017-03-15). Author: smccann@instructure.com. Available: <https://community.canvaslms.com/docs/DOC-9067-how-do-i-configure-live-events-for-canvas-data> Accessed: August 2018
- [13] Tableau. Tableau Home. Available: <https://www.tableau.com/> . Accessed: August 2018
- [14] Tableau. Tableau Workbook Examples. Available: <https://www.tableau.com/trial/tableau-examples> Accessed: September 2018.
- [15] Canvas Community. Document: Tableau Workbook Examples. (2016-08-21) Author: John Sheringer. Available: <https://community.canvaslms.com/docs/DOC-8121-tableau-workbook-examples> Accessed: August 2018.
- [16] European Commission. Data Protection Rules for business and organisations. Available: https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations_en Accessed: August 2018.
- [17] Kimball Group Website. A Dimensional Modelling Manifesto. Available: <https://www.kimballgroup.com/1997/08/a-dimensional-modeling-manifesto/> Accessed: August 2018.
- [18] Shneiderman, B., & Plaisant, C. (2003). *Designing the User Interface: Preview*. Addison Wesley.

Appendix 1

Attribute	Value description
id	Unique identifier for the event.
type	Type of event, for "asset_accessed" events this value is "NavigationEvent"
actor id	The Canvas id of the currently logged in user prefixed with "urn:instructure:canvas:user:"
actor type	The type of actor performing the event, either "Person" or "SoftwareApplication".
actor extensions entity id	The Canvas id of the currently logged in user.
actor extensions user login	The login of the current user (university network username)
actor extensions root account id	The Canvas id of the root account the user belongs to. (For students and staff, it is the ID of the university: "110220000000000001")
actor extensions root account lti guid	The Canvas Learning Tools interoperability ID of the root account of the current user.
edapp id	Usually "http://canvas.stir.ac.uk/" - changes only if it is an API call from Instructure
edapp type	"SoftwareApplication"
group id	The Canvas ID of the group prefixed with "urn:instructure:canvas:course:"
group type	"CourseOffering"
group extensions entity id	The Canvas ID of the group
group extensions context type	The type of context where the event happened.
action	"NavigatedTo"
object id	ID of the object that was navigated to, determined by the object name and asset type. E.g. If the object name is "assignment", then this field contains the Canvas ID of the assignment prefixed by "urn:instructure:canvas:course:"
object name	The name of the type of object navigated to, e.g. assignments, grades, course, etc
object type	"Entity"
object extensions entity id	Canvas ID of the object navigated to.
object extensions asset type	The type of asset being accessed e.g. "Course"

object extensions asset subtype	The item being accessed is identified by asset_type, asset_id, and asset_subtype. If asset_subtype is set, then it refers to a list of items in the asset. For example, if asset_type is course, and asset_subtype is quizzes, then this is referring to viewing the list of quizzes in the course. If asset_subtype is not set, then the access is on the asset described by asset_type and asset_id.
session id	The session identifier for this event.
session type	"Session"
context	http://purl.imsglobal.org/ctx/caliper/v1p1
eventTime	The time the event took place.
extensions version	1.0.0
extensions hostname	The hostname of the current request.
extensions request id	The identifier for this request.
extensions user agent	The user agent sent by the browser making the request.
membership id	Details about the user and the group page that was accessed, in the format: "urn:instructure:canvas:course:[entity ID for group being accessed]:Learner:[Canvas user id]" e.g: "urn:instructure:canvas:course:11022000000002670:Learner:1102200000001994"
membership type	If a group was accessed, this value is "Membership".
membership roles	The role of the current user in the current context. ["Learner"/"Instructor"]
membership member id	Unique canvas user id prefixed with "urn:instructure:canvas:user:"
membership member type	Usually "person", same as actor type
membership organisation id	entity id for object being accessed prefixed by urn:instructure:canvas:course:
membership organisation type	Usually "CourseOffering"
sensor	Usually "http://canvas.stir.ac.uk/"
sendtime	Time the request was sent.
dataversion	http://purl.imsglobal.org/ctx/caliper/v1p1