Predicting the Probability of Attendance of Hotel Guests

Christopher Philip Stewart

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
Software Engineering

School of Computing

04 2020

Authorship Declaration

I, Christopher Philip Stewart, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed: CStewart

Date: 03/04/2020

Matriculation no: 40281448

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

CStewart

The University may not make this dissertation available to others.

Abstract

Through research of the respective industry, it has been acknowledged that the hospitality trade, particularly the hotel industry, suffers greatly from the inability to guarantee that guests follow through with their room reservations. By implementing machine learning technology, alongside a simple to navigate graphical user interface, this paper details a proof of concept application, fit for use as a strong starting ground for development into further technological advances on this issue.

The proof of concept software comprises a Java-based application able to interact with an Artificial Neural Network, constructed in Python, to return the percentage probability of attendance regarding an individual booking. Further issues such as the necessity for the implementation of this software in the industry, and the governing bodies dealing with this issue are also discussed in detail, as well the interaction between humans and computers, and how this software has been designed to fit with both modern and historical Human Computer Interaction practices and guidelines.

Contents

List of Tables

List of Figures

Acknowledgements

I wish to acknowledge the help and guidance provided by Mr Simon Powers during the process of this dissertation.

1 Chapter 1 Introduction

In the hospitality industry, knowing your industry inside and out can reap large benefits for the business. From small, local businesses to large scale conglomerates, inside business knowledge is key to increasing sales and revenue. In days gone past, word of mouth and low-level relationships were how businesses spread their influence, however, in today's market, technology is a must in order to climb the rankings and stay on top of the marketplace.

Artificial Intelligence has been introduced to the hospitality trade, particularly in hotels, to help combat the lack of knowledge regarding customers and their likelihood of following through with their reservations. The aims of such software implementations focus on understanding the correlations between customers and the factors governing the occupancy of the hotel.

This dissertation amalgamates the background research into the hospitality trade's use of artificial intelligence, machine learning, and data mining to effectively understand the benefits and future implementations of such software in this field.

Secondly, the aim is to develop a proof of concept piece of software that will interpret data taken from a client involved in reservations and provide beneficial information to them, such as predictions and insights – in particular, the probability that an individual booking will be cancelled or the guests will not attend the reservation.

This software should provide a simple yet informative insight into a client's reservation data, input in the form of a comma-separated value file with the timespan of one month (i.e. a formatted spreadsheet containing the next month's room reservations), and present this information in an easy-to-understand manner. The software will utilise predictive data analysis techniques to deliver the ability to predict the impact of major changes to topics such as their market audience or the busiest times for reservations and will aim to discover the hidden relationships between variables in the booking data.

Variables such as the day, month, and year the booking was made, the number of adults attending, and the channels used when booking the room, to name a few, will all be entered into the machine learning algorithm in the software to train an accurate prediction model – using artificial intelligence to uncover hidden relationships between these factors and utilising these variables to predict the attendance probability of the booking.

Data will be taken from a sample data set made available to the public for educational and training purposes. This data has been collected over approximately 2 years and has been anonymised for data regulation compliance.

Attempts to include a corporate client for sounder data interpretation and utilisation have been pursued as far as possible, however, complications on the client's end have unfortunately resulted in this being impractical.

1.1 Main Deliverables

1. Easily understood front end software that will present the client with access to their predictions and insights as simply and as quickly as possible.

2. Back end software that competently and effectively runs algorithms on client data in a predictive analytical style.

1.2 Main Intermediate Steps

1. The application must provide useful and beneficial user feedback whenever an error or exception occurs.

2. Both applications must be able to communicate efficiently and effectively.

3. Any sensitive/personal information stored in a database must be correctly protected via use of appropriate hashing/encryption/etc. methods and calls to the database must be secured properly from injection.

4. A literature review composed of background research on the issues of predicting attendance in the hospitality industry.

5. A simple yet efficient and intuitive User Interface design.

2 Chapter 2 Literature Review

Hospitality is an industry that inherently possesses certain predictability issues. The market is competitive and those who strike the best deals with the most customers often make it out on top. Understanding how to beat the market before it changes grants a hotel, and any business for that matter, the ability to lead in sales and drive revenue far greater than trailing behind with second-hand knowledge.

With the industry being so focused on the customer's impression and satisfaction with the hotel and services, the risks involved for the company increases exponentially. Hotels are not in the fortunate situation where blame can be passed to suppliers and outside agencies as the hotel takes the brunt of the negativity caused by any issue.

One of the biggest risks involved here is vacant rooms. With a highly competitive market, hotels are expected to offer their services with the assumption that the customer will show up to receive said services. When the option to cancel the room is taken up by the customer, the hotel is left to face the consequences (Antonio, 2019). This can cause serious and significant impact to the hotel's ability to manage revenue-based decisions and thus the revenue itself.

The cancellation of a booking in a standard hotel accounts for approximately 20% of the hotel's total bookings and therefore is a substantial issue that faces the hospitality industry today (Romero Morales, 2010). With an issue in the hospitality field causing the net demand of a hotel to drop to just 71% within a 3 - 10 day timeframe from its due date, and with the airline hotel industry considering 60% of its booking to be cancellations (Antonio, 2019), the field is openly working towards finding a solution to this problem.

Some businesses now employ stringent cancellation policies in order to ward off the negative impact that this can have, however, in doing so, create another problem. By imposing rules and regulations on their guests, hotels may actually drive away business (Antonio, 2019). Tight restrictions on cancellations can heavily impact a hotel's reputation in the marketplace and, as such, should not be considered the ideal solution to this industry's problem.

In recent years, the use of data mining and forecasting techniques have been employed to counteract the knowledge gap between hotels and their occupancy rates. Accurate and well-developed forecasting measures are able to assist administrative and managerial teams carry out sound strategical marketing decisions that have the potential to make vast improvements to revenue and reputation operations (Tang, 2017).

The benefits that this can bring affects not only the smaller standalone businesses, such as local Bed & Breakfasts, optimise their occupancy rates by correctly assigning funds to where it matters, but grants large scale conglomerates the ability to predict overall tourism levels and make better decisions based on recent and accurate data.

(Tang, 2017) explains that due to the large costs involved in incorporating a data mining system into the infrastructure of a business, the option of using algorithms to forecast occupancy rates is kept out of reach for a lot of smaller scale hotels. The demand and necessity for a system such as this, however, is equally as high as it is for large businesses worldwide.

2.1 Data Mining and Forecasting

Data mining and forecasting consists of gathering a substantial amount of historical data that closely and accurately represents the daily intake and output of the hotel. This data is then cleaned of any “dirty” data – data consisting of inaccuracies, missing data, useless data. This data is then analysed through use of carefully selected algorithms that are run on a portion of the dataset, used as a training set, before the final portion of the data is used as the test set. The algorithm’s effectiveness and accuracy are then examined and evaluated before the most optimal algorithm is selected with the best parameters available.

In a study performed in 2014, Artificial Neuro Networks (ANNs) were implemented into the process of creating an effective forecasting software for the hospitality industry. ANNs are created with the intention of replicating the “cognitive capabilities and processing models of developed individuals” (Corazza, 2014).

An Artificial Neuro Network is a construct of hierarchical units, or layers, that divide up the processing of input data. This system replicates the manner in which the human brain processes information, with different sections of the brain dealing with its own part of the information. The data is passed through from, what could be, a few dozen layers, all the way to millions upon millions of increasingly important and more complex layers.

The Artificial Neuro Networks provide the algorithms a structured way with which to learn from data that they are receiving and processing. Returning to the training and testing datasets mentioned previously, ANNs provide the basis for this approach to data mining.

Predictive analysis, as opposed to descriptive and prescriptive analysis, is commonly adopted to solve this problem presented, with it either being approached as a regression problem or, less commonly, a classification problem. (Antonio, 2019) In keeping with the predictive analysis approach to this machine learning problem, the researchers state that the model built is to comprise two individual sections. Firstly, the machine learning algorithms and networks required to process the historical data input to the software and provide predictions on new/unseen data in future. Secondly, a separate method of evaluating the predictive capabilities possessed by the models. (Antonio, 2019)

When the approach to the solution is finalised, data then becomes of utmost importance, requiring solid background and reliable sources. When collecting data on this scale, huge efforts are required in order to secure complete and trustworthy data. It is here that global organisations come into play, supplying researchers and data analysts with the information required in order to study and build upon data collected from worldwide sources.

2.2 The OECD

The Organisation for Economic Co-Operation and Development (OECD) is an international organisation that has collected vast amounts of data regarding the hospitality trade and provides this data openly to the public for use in business. Currently, 36 countries are active members of the organisation, feeding their data into a massive international collection that is then used to create a sounder platform for global-scale decision making, identifying and finding solutions to commonly encountered problems, and the coordination and evaluation of policies.

The OECD has defined a few indicators that can are described as “the OECD composite leading indicator (CLI; combined economic variables that indicate a country’s economic situation), the OECD business survey index (BSI; a qualitative survey with top managers who possess national expertise) and the OECD consumer confidence index (CCI; quantitative research undertaken with local citizens about their living standard).” (Tang, 2017)

To optimise the effectiveness of machine learning in a hotel environment, the OECD data can be used alongside the data already collected by the hotel itself in order to provide far larger, albeit more general, datasets for training and testing.

The use of this information can be effective in enabling stronger predictive technologies in hotels, however, also provide businesses with a direct link to the greater marketplace (Tang, 2017). A hotel may use this data, along with data mining and forecasting methodologies, to greater understand the global and regional effects in the industry, allowing for more practical and effective decision making throughout the business and/or its branches.

(Tang, 2017) describes that businesses stand a far greater chance of having stability in the market by aggregating multiple sources of information from both micro and macro-environmental domains. These domains differ from each other, with the micro environmental data proceeding from the internal running of the business – regarding staff, hotel cleanliness, and general satisfaction of the services provided – whereas the macro environmental data originating from global trades and markets, tourism influences, and large scale economic factors that may be out-with the smaller businesses control. Understanding the market and predicting future occupancy rates may benefit significantly from a sound, encompassing dataset attributed to multiple, varied yet reliable sources.

2.3 Human Computer Interactions

With so much emphasis on datasets and machine learning, it may become easy to forget about the physical employee in these situations. Human Computer Interactions (HCI) focuses on the relationship between physical users and their technology, examining how, when, and why users do and feel what they do. With user requirements and expectations at the forefront of most software nowadays, with development teams favouring agile methodologies over more traditional waterfall methods, HCI is an ever adapting field that is constantly being updated.(Lallemand, Gronier, & Koenig, 2015)

The term “User Experience” was first coined in the 1990’s by a small group of authors, each individually introducing the phrase in texts. It was believed that “usability” was a term that failed to describe every aspect of a user’s interactions with technology. During the same period of time, the term “quality of experience” was introduced, focussing on the user’s thoughts, feelings, level of achievement felt, and the context in which the entire interaction took place.

Whereas the software and implementation of the data mining techniques employed are dependant on the skill of the engineers, user experience can only truly be affected by the client companies. It can be said that the key stakeholders in these projects actually own the user experience due to a few factors:

- When software is complete, user experience heavily impacts the market audiences’ reaction to the product produced.
- The importance of bringing high-quality user experience products to a rapidly evolving marketplace is understood greatly by the stakeholders
- Stakeholders possess the power to structure organisations in a manner that collectively contributes to the team based efforts in creating a great user experience.(Bogaards & Priester, 2005)

Usability and user experience are still colloquially referred to in a synonymous manner, prompting a study that looked at the perception of what user experience actually means in the workplace. It was found that the understanding of user experience has adapted through the years to closer to define HCI principles in the ever changing marketplace. Respondents to the study agreed more that “usability is a necessary precondition for good UX” than in a study performed in 2006. To add to this discovery, the study showed that the statement “UX is based on how a person perceives the characteristics of an artefact, but not on the characteristics per se” had become less relevant and accurate in the industry, dropping 7 places in ranked statements from 2008 to 2012. (Lallemand et al., 2015)

HCI, however, does not solely focus on the user’s interpretation of the software based on looks and design. Artificial Intelligence (AI) has struggled for many years to become a trusted tool for many businesses. Error rates in machine learning algorithms and implied sexual biases are among the many unwanted factors creating a poor relationship between AI and companies willingness towards its implementation.

To combat this, several solutions have been proposed to increase the trust in artificial intelligence-based software. Notably, the likening of this kind of software product to other trades, such as electrical engineering and medicinal practices, industries where documentation is required by law with strict guidelines and standardised layouts, prompted the proposal of datasheet for datasets.(Gebru et al., 2019)

Gebru et al. also continue with this proposal to discuss the standardised layout and information provided by such a document. They split the datasheet into extensive categories, with each heading containing the answers to multiple background questions.

These categories are said to consist of:

- Motivation
- Composition
- Collection Process
- Preprocessing / Cleaning / Labeling
- Uses
- Distribution
- Maintenance

It should be noted that Gebru et al. state that the questions and headings included in this list are subject to change as datasheets become more widely used and readily available in machine learning and AI software practices.

A slightly different approach to this proposal for a industry-wide adoption of standardised datasheets, is the CERT Secure Coding Standards (CSCS). This is a set of coding standards implemented by the Software Engineering Institute that details the common flaws and consequences of flawed and even compliant code. (Arnold et al., 2019) The CSCS provide auditing tools capable of scouring code to identify and examine any breaches of the secure standards.

The need for a tool worthy of this work is made essential by the sheer number of rules and recommendations imposed on programmers - with 118 and 182 respectively, just for the C programming language.(Olesen, Hansen, Lawall, & Palix, 2010) The cited also discuss the unruly nature of C and how, with such a vastly unrestrained language, programmers themselves can easily fall victim to unknowingly developing loopholes in software, rendering the entire application unsafe and sometimes even dangerous.

Using past experience of attacks on large scale software, leading businesses perform post-deployment remediation of vulnerabilities. This can be described by looking at Microsoft's rolling update scheme. Every second Tuesday, an update patch is released in an attempt to thwart potential attacks by consistently fixing bugs and errors in system security. However, the United States Department of Defence (DoD) declare this as an "insufficient and expensive means of securing of securing networked systems".('Supporting the Use of CERT® Secure Coding Standards in DoD Acquisitions', n.d.) Organisations and companies whom rely heavily on the software capabilities provided to them must have absolute trust in the security and accuracy of the product.

With this being the case, DoD released Directive 8500.1E on information assurance. To quote, the directive "establishes policy and assigns responsibilities to achieve DoD information assurance (IA) through a defense-indepth approach that integrates the capabilities of personnel, operations, and technology, and supports evolution to network centric warfare" ('Supporting the Use of CERT® Secure Coding Standards in DoD Acquisitions', n.d.)

Directives like these provide top-of-the-line assurance to businesses and clients alike, increasing the likelihood of trust building in AI operated software. With such large government departments issuing well examined directives, international certification authorities releasing guidelines and rules regarding secure and standardised coding practices, and continuing research and propositions of industry-wide, standardised datasheets and background information releases, HCI is a consistently adapting field working to improve the relationship between technology and humans.

Even with directives and datasheets implemented into the standard workflow, there is an overriding, inherent stigma around the use of artificial intelligence and forecasting. For years, an aversion to the advice given and information delivered from a computer-based algorithm has been present in humans, particularly when pitted against advice that has come from another person. (Prahl & Swol, 2017)

This effect has been dubbed “algorithm aversion”. Research into this psychological effect is ongoing, however, the vast number of studies performed utilise anecdotal experiences instead of concrete, real life data. (Dietvorst, Simmons, & Massey, 2015)This has prompted these researchers to investigate this effect with empirical data to produce previously unseen factual results.

In running this study, it was concluded that even when an algorithm produces half as many errors when forecasting situations, the human option is preferred - this effect was highly significant in all tests performed. This is believed to be due to the participants being exposed to the algorithm, witnessing it work, and therefore make errors along the way of its learning. (Dietvorst et al., 2015)

(Castelo, Bos, & Lehmann, 2019) describes an argument to explain the reason for this effect, claiming that it may be due to the innate desire to shift blame from oneself onto another. This paper concatenates the theories that “humans, unlike most algorithms, can provide explanations for their decisions, are perceived to have high confidence, have a reputation to maintain, and have information about future events”.

It appears that humans have a far greater threshold of tolerance for other humans and their mistakes than they do for machines and theirs. The impact of receiving poor advice from both a human and a machine differed significantly in a study performed in 2016. The trust in the algorithm presented to participants severely degraded when the machine made an error in its prediction, however, a much lesser decline in trust towards the human was observed. (Prahl & Swol, 2017)

The resounding truth with AI and machine learning and forecasting is that, for every success achieved by the advancement of technology, human conditioning and nature unconsciously provides psychological barriers that cannot be avoided in this industry. HCI therefore takes an incredibly imperative role in delivering a piece of software that not only works in terms of business case achievements, but is compatible enough to become an integrated part of human’s workflow and daily life without raising too great an issue with innate human preconceptions. (Dietvorst et al., 2015)

The components of solid HCI based software vary depending on the context of the software. During a comparative study, it was observed that the most significant factors influencing a user’s experience, notably the visibility of system status, interactivity, and learning design, completely changed when evaluating software in a different context. In the compared software, the heuristic factors were deemed to be variables were accessibility, navigability, user control, error tolerance and flexibility, and the readability and quality of the writing. (Rutter, M. J., Smith, S., & Alshehri, A., 2019)

With HCI’s value variables changing with each application context, a gap in the understanding of HCI has been identified, with designers often assuming the importance of different heuristics than those perceived by the user. In one study, the “easy to learn” heuristic - categorised as an effectiveness characteristic - was rated higher by software developers whilst “privacy” - deemed as an emotional characteristic - was rated higher by users. (Cata & Martz, 2015)This brought researchers to the conclusion that designers may be underestimating the impact and value of emotional characteristics when using products.

With the aim of resolving this issue, usability design principles were released by Gould and Lewis. Though early in the technology field’s lifetime, they had deemed four design principles to be generic yet applicable throughout HCI design. These four principles being an early focus on users and tasks, gaining empirical measurements through scenarios and prototypes, following an iterative design structure, and finally integrating usability to become one aspect under control as a whole.(Gould & Lewis, 1985)

3 Chapter 3 Application Design

With consideration of software engineering best practices, the application will be split into two main pieces of software. These will consist of a UI based application, built using Java and the Swing WindowBuilder libraries, and a machine learning model constructed using Python and the Keras and Tensorflow functionalities.

The Java programming language has been selected due to its deep integration in the modern technological industry, its Object-Oriented Programming fundamentals, and the vast number of readily available libraries and rich APIs.

As Python is quickly becoming the go-to machine learning language, as well as the extensive plethora of documentation and support for libraries such as the aforementioned, it was quickly, yet consciously, decided upon as the language of choice for the machine learning module.

Here, the architectural design of the Java application will be discussed, and henceforth will be known as the “Predictor” application for ease of naming.

3.1 Java Application - Predictor

The Predictor application has been designed in a three-tier architectural format consisting of an Application layer, a Business Layer, and a Data Layer. The naming convention of these layers can vary and so it is worth noting that when carrying out further reading, the layout of this architecture is synonymous with a Presentation Layer, followed by an Application Layer, and finally a Data Layer.

The separation of code through use of these layers allows for a far smoother adoption of new features and adaptation of current logic. The Application Layer’s code is to consist of everything front-end, readily available to the end user for interaction, with only calls to the next layer to request outputs and/or functionality to take place. In doing so, the application is not only far safer in terms of malicious code manipulation from malevolent sources, but also greatly reduces the amount of code needed in order to tweak functionality already implemented within the layers.

The Business Layer will contain all code specific to the interpretation and manipulation of data, providing the main application functionality. This layer will sit between the user accessible Application Layer and the safeguarded Data Layer, acting as a middleman between the two. Within this layer will lie the classes required in order to perform such actions as IO file access, user login, and of course, bridging the gap between the java application and the Python machine learning script. This architecture means that the Data Layer has nothing but its own functionality to focus on - the persisting, retrieval, and updating of data in the database.

Should the maintenance of this application ever come under the responsibility of a team of developers - as opposed to one individual - then no longer would a full stack developer be required, and as such, the workload could be divided up easily to certain stack specialists.

3.2 Python Script

The machine learning algorithm will be developed using Python and the Keras and Tensorflow libraries. It will involve the creation of a deep learning neural network, to be trained on the approximate 78,920 booking reservations collected from a study[1] displayed on ScienceDirect, performed between the 1st of July 2015 and the 31st of August 2017.

The dataset consists of 31 variables, a mix of integer and categorical data. The fields, along with their data types are listed below.

Table 1 Dataset Variables and Types

Variable	Data Type
IsCanceled	Integer
LeadTime	Integer
ArrivalDateYear	Integer
ArrivalDateMonth	Integer
ArrivalDateWeekNumber	Integer
ArrivalDateDayOfMonth	Integer
StaysInWeekendNights	Integer
StaysInWeekNights	Integer
Adults	Integer
Children	Integer
Babies	Integer
Meal	Categorical
Country	Categorical
MarketSegment	Categorical
DistributionChannel	Categorical
IsRepeatedGuest	Integer
PreviousCancellations	Integer
PreviousBookingsNotCanceled	Integer
ReservedRoomType	Categorical
AssignedRoomType	Categorical
BookingChanges	Integer
DepositType	Categorical
Agent	Integer
Company	Integer
DaysInWaitingList	Integer
CustomerType	Categorical

[1] https://www.sciencedirect.com/science/article/pii/S2352340918315191

ADR	Numeric
RequiredCarParkingSpaces	Integer
TotalOfSpecialRequests	Integer
ReservationStatus	Categorical
ReservationStatusDate	Date

The structure of the Python script that will be made available for end users is planned to be that of a lightweight piece of software that will load a prebuilt model, before assigning weights and conducting the predictions. The actual model will be constructed in a separate script, where it will be possible to conduct analysis on the neural network to determine the most optimal hyperparameters and plot these statistics for developer review. As neither the end user nor the application have any need to consistently build the model from scratch, the best design stage plan is to remove this unnecessary load.

4 Chapter 4 Implementation and Results

4.1 Python Script Implementation

Before beginning this project, a solid understanding and firm knowledge base of machine learning algorithms were lacking. Through research and study of common machine learning libraries and languages, the following arrangement of technologies were implemented:

- Keras
- Tensorflow
- Python

The python script uses the Pandas library's functionality to read in data in csv format, perform a series of steps to manipulate the data into a valid format for inputting into the model, and then predicts the probability of guests showing up or not for the individual bookings.

The dataset provided by the cited study included several instances of dirty data. The removal of this dirty data is well worth noting as the machine learning model used in the Python script relies on the data being valid before inputting into the script. The validation of this data is discussed further later on, where user input validation in the Predictor application is described. The steps taken to remove or remedy these entries are as follows:

- Entries with no adults were removed
- Trailing and leading white spaces were removed from several columns
- Entries will NULL as their Country value were removed
- Entries with an N/A value for Children were replaced for 0 in this column
- Undefined values within the Market Segment and Distribution Channel columns were omitted

Graphs depicting the effects that varying hyperparameters had on the accuracy of the model were plotted in order to best determine the most optimal configuration. The model was compiled 30 times, each time having the validation accuracy of the *"fit()*" function stored in a list, allowing the mean accuracy for those 30 runs to be calculated.

This process was applied to the number of neurons within the dense layers of the model as well as the batch size used. These graphs were imperative in concluding the final configuration – a total of 33 neurons (16, 16, 1), a batch size of 32, and 10 epochs. The model also utilises early stopping in order to minimise overfitting as much as possible, allowing the model to stop training and validating during an epoch should the validation loss start to increase again.

The Java-based application is able to create a CMD (Command line) command to execute the Python script. It passes the necessary arguments, such as the location of the training file and the file presented by the user, and retrieves the output from the script - which is then read back into the Java application and processed before amalgamating it with the user's input data. (Fig.11)

4.2 JDBC and WampServer

As the application has been developed within the confines of a "proof of concept" model, in order to demonstrate the feasibility, as well as the operational benefit to be gained from such an expenditure, the Predictor application makes use of some readily available software and libraries in order to assist its development.

JDBC - **J**ava **D**ata**b**ase **C**onnectivity - is a java-based API that allows the determination of how a client may access a remote database. This API satisfies the needs of the Predictor application well, is readily available, and is widely used and well documented. Though alternatives do exist, and in certain situations provide faster functionality and/or easier maintainability in future, the necessity for this has not had a significant impact on the implementation of JDBC within this application, and such, JDBC has been utilised to provide database connectivity and functionality.

WampServer - **W**indows, **A**pache, **M**ySQL, **P**HP - is a bundled software stack created to facilitate the use of an application's connection to a database in a safe and secure environment, namely a development of prototyping project. It is understood that the end user company utilising the Predictor application will have a database in place with predefined dependencies and software restrictions. As this project's aim has been to develop a proof of concept application, the use of WampServer has been adopted in order to create a functional piece of software able to demonstrate the appropriate use cases without venturing too far in one direction, potentially straying from certain hotel infrastructures.

4.3 IDE and Version Control Software

The integrated development environment (IDE) selected for the development of both the Java and Python applications was that of Eclipse. Regarding the Java-based Predictor application, the WindowBuilder plug-in has been used in Eclipse to provide graphical user interface (GUI) development functionality. WindowBuilder bundles SWTDesigner and Swing Designer to generate a layout capable of handling varied application use cases. It is this that provides access to the windows and other GUI used within the application without the need for additional library compilation and can be used without WindowBuilder installed after deployment.

In order to develop the Python-based script, PyDev for Eclipse was utilised. PyDev provides all of the expected and necessary development tools required for developing Python scripts within the Eclipse environment, allowing for the construction of two seperate language-based applications to take place within the same IDE. This also increased the effectiveness of the use of a version control software (VCS).

EGit was implemented into the Eclipse development environment in order to allow connection to the VCS, GitHub. It is here that control over versions of the software was held, with two main branches in use - the master branch and the development branch. The master branch, also known as the live branch, contains only the working software version as its head node. As features and other additions or edits are made to the code, nodes are pulled into the development branch.
As this project has been developed by one individual with sole ownership of any and all code implementation, only the development branch exists alongside the master. This, however, would not replicate well within a team of developers and, as such, further branches should be pulled per feature from the development branch to avoid confusion between commits and developers stepping on each other's toes, as it were.

4.4 HCI Integration and User Experience

It is understood that the technical competence and experience level of the end user is assumed to be less than that of the programmer or a technically knowledgeable individual to any advanced degree. It is therefore assumed that the application is susceptible to a wider range of human error. Due to this factor, as well as the basic software engineering principles regarding the baseline of user experience (UX), measures have been taken in order to prevent fatal errors from occurring during the use of the program.

Firstly, the user experience and HCI integration in the application will be discussed with a view to no expected malicious intent. This means that it is assumed, in this following section, that any error-making and behaviour outwith that expected is simple human error and is not deemed to be an attack on the software or surrounding data sources.

To combat simple errors in user defined inputs, such as during login and defining database authentication details, input validation has been implemented to catch values outwith that expected and return useful feedback to the user. Upon starting the application, a check is performed for a properties file, within which are the username, password, and url required for connecting to the database. Should this file not exist in the required location, a message is displayed to the user explaining the need for the file and directs them to an interface where the file is created, before returning them to the login screen.

Now acknowledging that the robustness and integrity of the application is paramount to the protection of users and their data, security measures have also been implemented throughout the application in order to safeguard from attack. To protect users and their respective companies, the user authentication data stored in the database has been salted and hashed using the BCrypt library to ensure the security of account passwords (Fig.5). When querying this database for said data, the application utilises prepared statements in order to defend itself from an SQL injection attack and ensure only predefined queries are performed.

The details required for connecting to the database are stored in a properties file, for use later in the application. These details are entered at setup and need not be re-entered again (fig.2). Should this file be removed at any stage during the application's running however, or if the authentication details entered are invalid or incorrect, then the next time the file is called upon, the application will display an error dialog explaining the issue and will redirect the user to complete the setup of this information again before continuing.

Finally, when inputting the file path for the bookings file, the user is presented with a file picker dialog (Fig.8), where the type of the file and its contents are checked for errors before it is passed to the machine learning model. The final prediction is output alongside the inputted data in a table provided in the application. Upon request, the user may save this prediction. In order to prevent vast amounts of data being transferred to the database, the decision has been made to save the file locally and instead save the overall statistics of the prediction to the database, available for later retrieval. Images of the user feedback, error handling, and expected use of the software can be seen in the extended figures section attached at the end of the paper.

4.5 User Experience Testing Plan

It was the initial plan for this software project to involve potential end users in a user experience testing exercise. Users were to be brought individually in front of the application and assessments were to be made on factors such as, for example, how long it took to complete a task, or how intuitive the application felt to the user.

Users were to be provided with forms allowing them to detail their experience, noting the positives and the negatives about the software, where this data would then be collated and reviewed for further development opportunities.

Due to the unexpected constraints brought forward by the COVID-19 virus, these in-person user experience testing sessions were therefore cancelled following government advice and remain outstanding for this software.

Provided below is the outline plan for the User Experience (UX) testing.

Number of Users to be involved: 2

Due to the nature of the application, only a couple of users would be necessary to test the application. This is also influenced by time and resource constraints that may be imposed on the client due to the daily running of their business.

User Description: Administrative user in the hospitality industry. Working knowledge of computer software related to the trade. User will be representative of the typical employee tasked with using the software.

Expected Time: 5 minutes

Goals of Testing:

1. Ensure users are able to intuitively manoeuvre through the application.
2. Users are comfortable with understanding and utilising the onscreen feedback
3. Users feel satisfied with the functionality provided to them when using the application

Format of Study: In-person.

This will allow for far greater understanding of the user's concerns and/or feedback and provides for a more robust testing of the software's usage.

Task 1: Sign in using the username: *admin* and password: *admin*

Task 2: Change the authentication details for the database connection

Task 3: A spreadsheet file of reservations has been provided. Use the software to predict the attendance rate of these bookings.

Task 4: Save the prediction. Now find the statistics provided about the most recent run.

Answer Format:

Task X

How simple was it to complete this task? Please circle one answer.

Impossible Difficult Average Straight forward

Did you come across any technical issues? Y/N

If you encountered any error messages, did you find these useful? Y/N N/A

If you have any other feedback, please provide this below:

5 Chapter 5 Evaluation

Upon evaluation of the entire process of this dissertation, it can be concluded that the aims set out previously have been achieved and a working, proof of concept application has been delivered. The Java application consistently and concisely produces the aimed for results whilst managing to cope with user error and technological exceptions.

The software implements security in several fashions – from securing user details in a database, using salting and hashing methods, to detecting faults in user setup and usage. The Predictor application effectively communicates with the Python machine learning script to generate consistent predictions when requested.

It is therefore made clear that the design and implementation used throughout this process can be brought to a client in the hospitality industry, at varying levels of market share and business development, and produce a working piece of bespoke software.

5.1 Project Management

Project deadlines and predicted timelines were clearly marked out during the beginning start-up phase of this dissertation. Liaison with the project supervisor took place in the form of weekly meetings in which the aims, deliverables, and scope of the project were discussed and defined, before transitioning into weekly progress goals (fig. 17). The Initial Project Overview (IPO) has been appended to the dissertation to demonstrate the planning and consideration that took place prior to beginning research.

The level of project management required for this dissertation was significantly higher than that previously necessary for other works experienced and as such proved slightly harder to maintain throughout the entire length of the process. With this being said, however, the constant weekly communications with the project supervisor proved invaluable in maintaining a dedicated focus to the incremental developments achieved.

Predicted timelines for objectives were able to be adhered to during the initial weeks with only minor time delays, most of which were no more than a couple of days. However, when implementation of the code was started, significant time delays were experienced due to debugging. This, however, did not cause a serious or harmful knock-on effect as the plan allowed for smaller tasks, such as the UI planning and design stages, to work alongside the implementation of code should it be necessary.

During the latter stages of the project, severe and unprecedented actions were exercised due to the COVID-19 pandemic. This heavily impacted the final processes, as evidenced in the Gantt chart provided (fig. 18). In-person weekly meetings with the project supervisor were rescheduled and performed via video link and email communications.

5.2 Improvements and Further Development

Having completed the project, areas for improvement, should the project ever be reconstructed, have been acknowledged. I am content with the learning achieved through development of this software as the initial aim was to engineer an application with which the level of experience with the associated libraries and languages had a solid baseline but the degree or manner of the deliverables went beyond that of something previously performed. The intention here is that the Honours project should provide grounds to showcase the ability to learn and adapt to new technologies when and where possible.

Due to first-time debugging of aspects of the application, workarounds and/or scaling back of certain features were implemented to achieve a working piece of software within the allotted timeframe.

Firstly, the machine learning script provided great difficulty when trying to load the saved model into the lightweight script intended for the user by displaying an error with which extremely little documentation was provided. As a result of this, the model rebuilds itself before performing a session of predictions in order to work through this issue and therefore provides a good starting point for possible future improvement – to further slimline this process with a saving and loading procedure for the model.

Several attempts were made, taking varying routes, to achieve a fully secured method of encrypting the properties file adequately. With the knowledge that the application should never store the encryption/decryption key within the source code, investigation was performed into using a master key within an online key store such as Google’s Cloud Key Management Service (KMS). Though the intentions and procedures of this are understood, issues in debugging this process proved too costly in regards of time management and therefore should certainly be a priority in further developing the software.

References

Arnold, M., Bellamy, R. K. E., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., … Varshney, K. R. (2019). FactSheets: Increasing Trust in AI Services through Supplier's Declarations of Conformity. *ArXiv:1808.07261 [Cs]*. Retrieved from http://arxiv.org/abs/1808.07261

Bogaards, P., & Priester, R. (2005). User experience. *Interactions*, *12*(3), 23. https://doi.org/10.1145/1060189.1060209

Castelo, N., Bos, M. W., & Lehmann, D. R. (2019). Task-Dependent Algorithm Aversion. *Journal of Marketing Research*, *56*(5), 809–825. https://doi.org/10.1177/0022243719851788

Cata, T., & Martz, B. (2015). Comparing Mobile APPs Usability Characteristics for Designers and Users. *Journal of International Technology and Information Management; San Bernadino*, *24*(4), 63–78.

Dietvorst, B. J., Simmons, J. P., & Massey, C. (2015). Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, *144*(1), 114–126. https://doi.org/10.1037/xge0000033

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumeé III, H., & Crawford, K. (2019). Datasheets for Datasets. *ArXiv:1803.09010 [Cs]*. Retrieved from http://arxiv.org/abs/1803.09010

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, *28*(3), 300–311. https://doi.org/10.1145/3166.3170

Lallemand, C., Gronier, G., & Koenig, V. (2015). User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. *Computers in Human Behavior*, *43*, 35–48. https://doi.org/10.1016/j.chb.2014.10.048

Olesen, M. C., Hansen, R. R., Lawall, J. L., & Palix, N. (2010). Clang and Coccinelle: Synergising program analysis tools for CERT C Secure Coding Standard certification. *Electronic Communications of the EASST*, *33*(0). https://doi.org/10.14279/tuj.eceasst.33.455

Prahl, A., & Swol, L. V. (2017). Understanding algorithm aversion: When is advice from automation discounted? *Journal of Forecasting*, *36*(6), 691–702. https://doi.org/10.1002/for.2464

Rutter, M. J., Smith, S., & Alshehri, A. (2019). Assessing the Relative Importance of an E-learning system's Usability Design Characteristics Based on Students' Preferences. *European Journal of Educational Research*, *8*(3). https://doi.org/10.12973/eu-jer.8.3.839

Supporting the Use of CERT® Secure Coding Standards in DoD Acquisitions. (n.d.). Retrieved 30 October 2019, from https://figshare.com/articles/Supporting_the_Use_of_CERT_Secure_Coding_Standards_in_DoD_Acquisitions/6584468

Antonio, N. A. (2019). An Automated Machine Learning Based Decision Support System to Predict Hotel Booking Cancellations. *Data Science Journal*, 18(1), 32. Retrieved from https://doi.org/10.5334/dsj-2019-032

Corazza, M. F. (2014). An Artificial Neural Network-based Technique for On-line Hotel Booking. *Procedia Economics and Finance*, 45-55.

Romero Morales, D. &. (2010). Forecasting cancellation rates for services booking revenue management using data mining. *European Journal of Operational Research*, 554–562.

Tang, C. M. (2017). Predicting hotel occupancies with public data: An application of OECD indices as leading indicators. *Tourism Economics*, 1096–1113.

Appendix 1 Figures

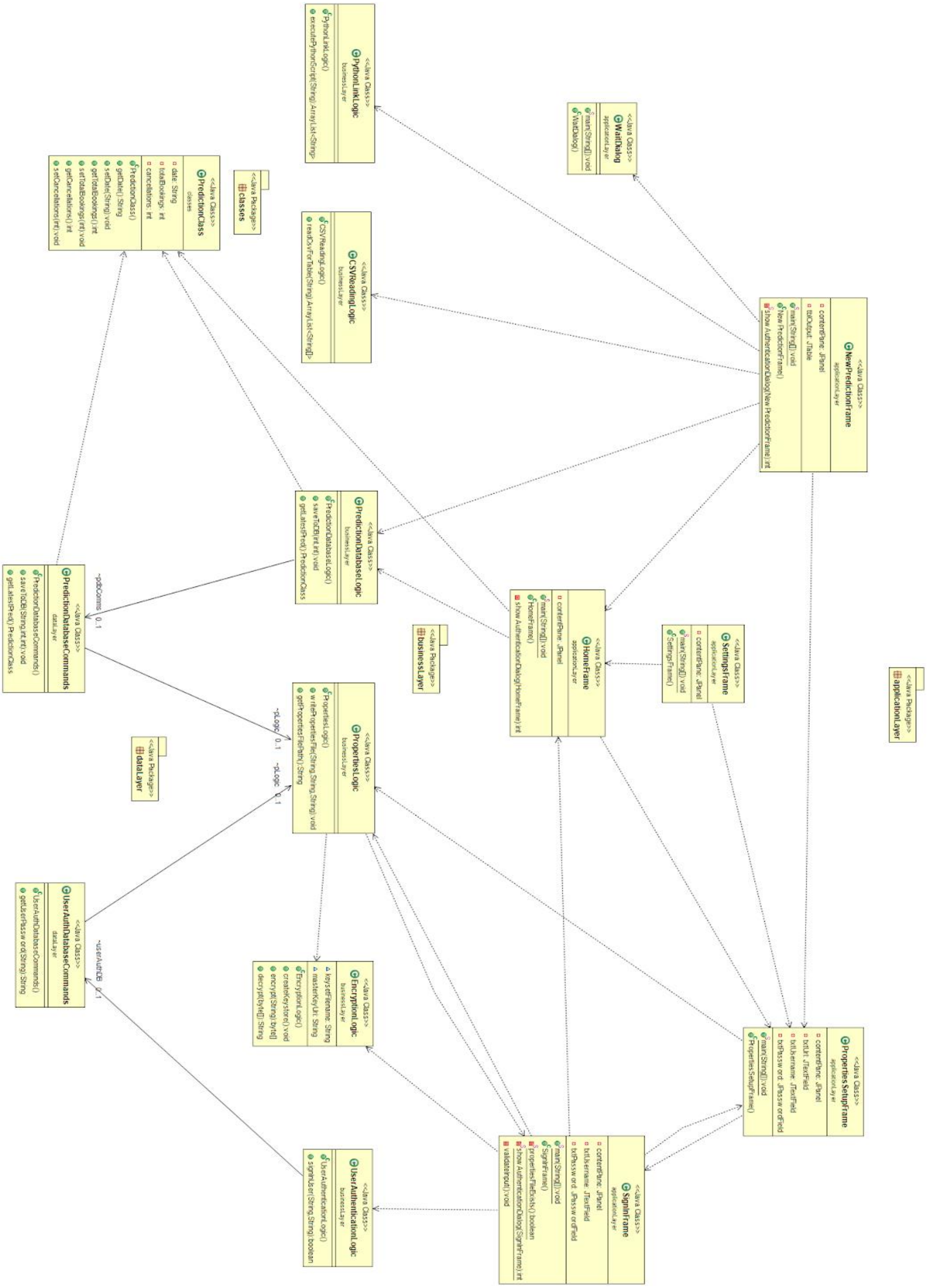
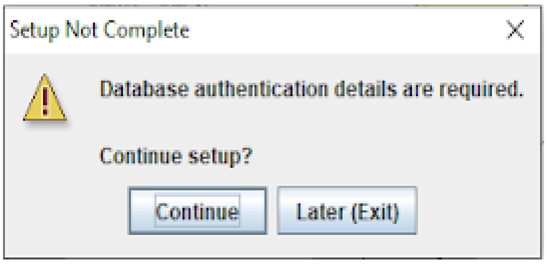


Figure 1 Class Diagram



Database authentication details are required.

Continue setup?

Figure 2 Setup Incomplete Error Handling

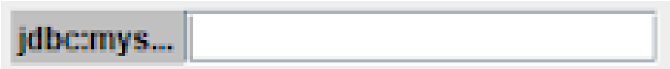
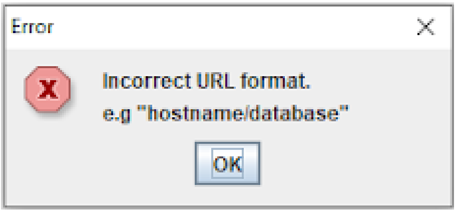
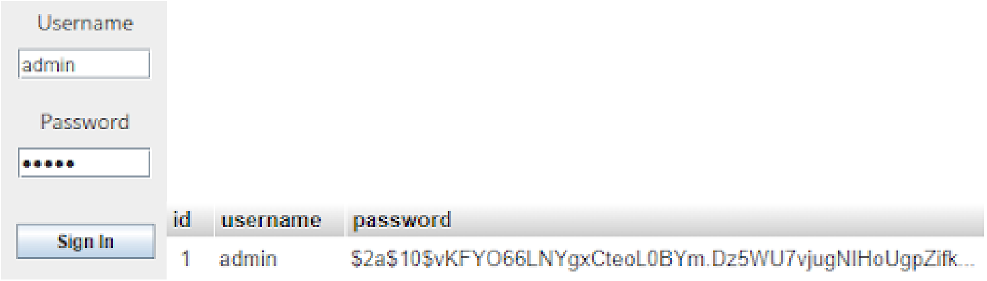


Figure 3 User only enters the final part of the address to minimise error



Incorrect URL format.
e.g "hostname/database"

Figure 4 Useful feedback after Regex returns error during input validation

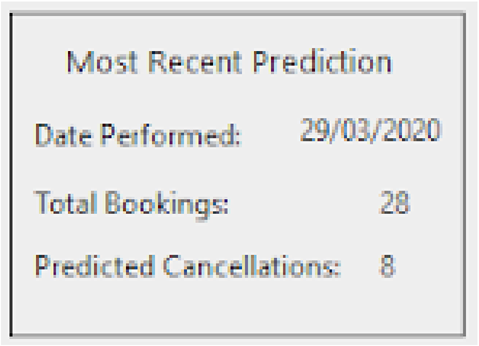


Username: admin

Password: •••••

id	username	password
1	admin	$2a$10$vKFYO66LNYgxCteoL0BYm.Dz5WU7vjugNlHoUgpZifk...

Figure 5 Passwords are salted and hashed for storage



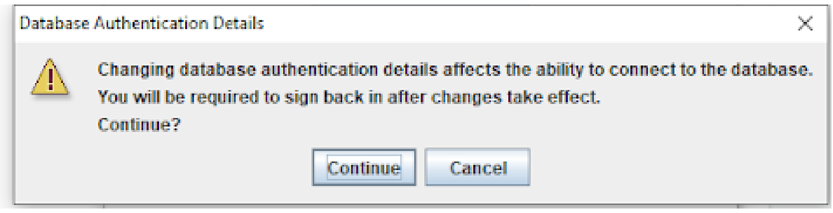
Most Recent Prediction

Date Performed: 29/03/2020

Total Bookings: 28

Predicted Cancellations: 8

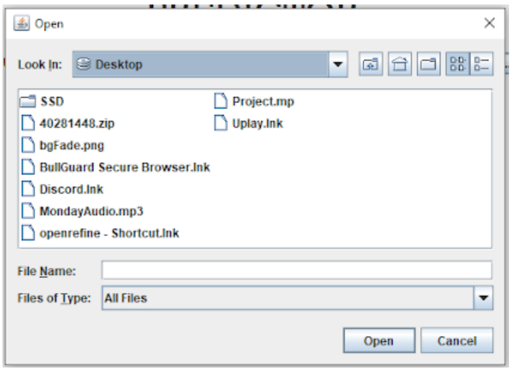
Figure 6 Most recent prediction statistics recalled for quick viewing



Database Authentication Details

Changing database authentication details affects the ability to connect to the database.
You will be required to sign back in after changes take effect.
Continue?

Figure 7 Warning message before user attempts to change database authentication details

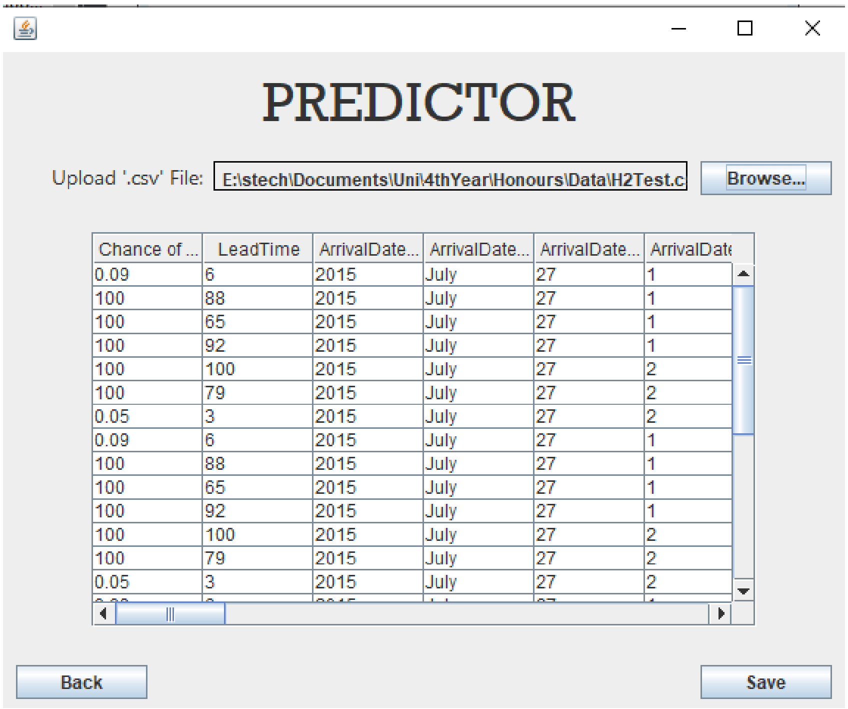


SSD
40281448.zip
bgFade.png
BullGuard Secure Browser.lnk
Discord.lnk
MondayAudio.mp3
openrefine - Shortcut.lnk
Project.mp
Uplay.lnk

File Name:

Files of Type: All Files

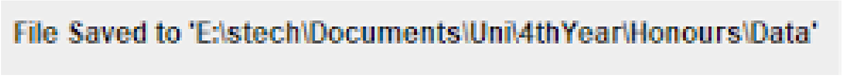
Figure 8 File Picker replaces the need for manually inputting file path



Upload '.csv' File: E:\stech\Documents\Uni\4thYear\Honours\Data\H2Test.c

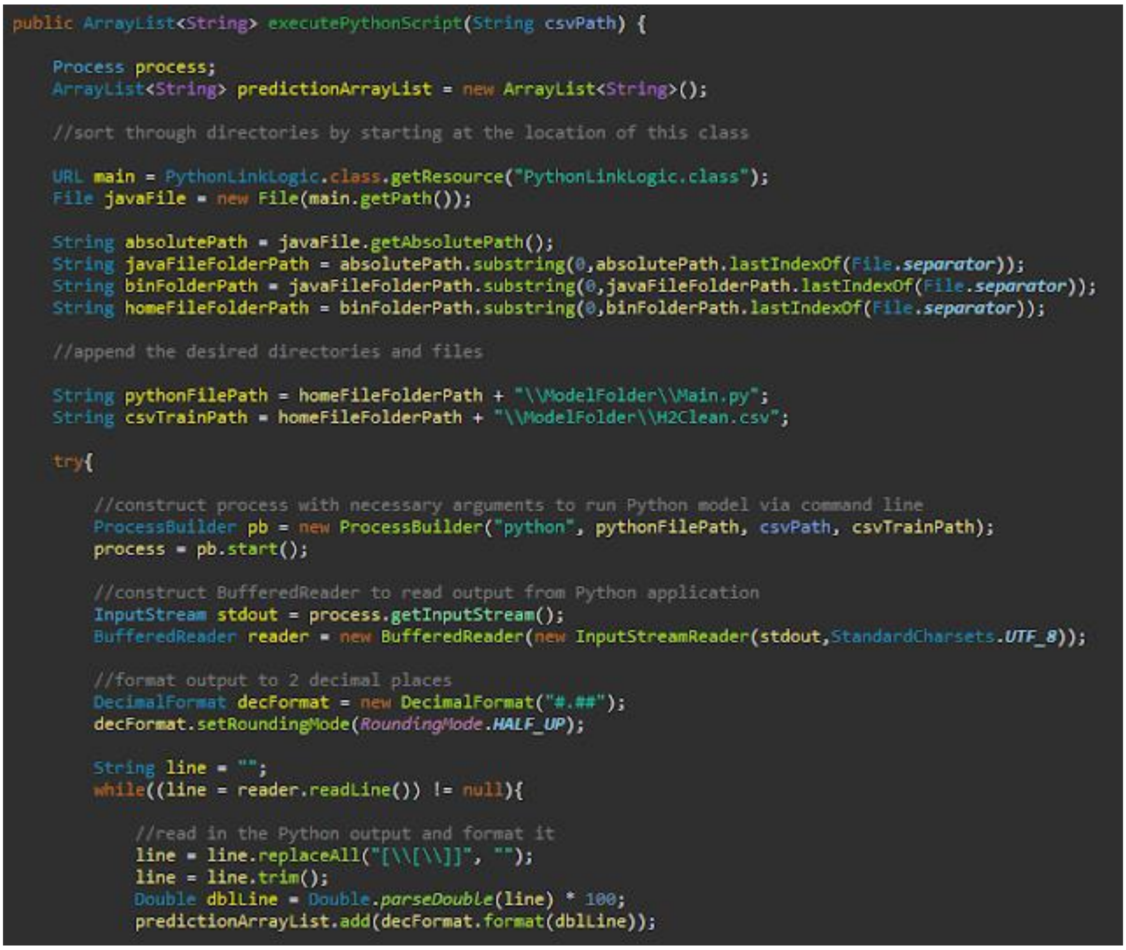
Chance of ...	LeadTime	ArrivalDate...	ArrivalDate...	ArrivalDate...	ArrivalDat
0.09	6	2015	July	27	1
100	88	2015	July	27	1
100	65	2015	July	27	1
100	92	2015	July	27	1
100	100	2015	July	27	2
100	79	2015	July	27	2
0.05	3	2015	July	27	2
0.09	6	2015	July	27	1
100	88	2015	July	27	1
100	65	2015	July	27	1
100	92	2015	July	27	1
100	100	2015	July	27	2
100	79	2015	July	27	2
0.05	3	2015	July	27	2

Figure 9 Prediction output to table alongside the input file's data



File Saved to 'E:\stech\Documents\Uni\4thYear\Honours\Data'

Figure 10 User is provided with the path of saved file



```java
public ArrayList<String> executePythonScript(String csvPath) {

    Process process;
    ArrayList<String> predictionArrayList = new ArrayList<String>();

    //sort through directories by starting at the location of this class

    URL main = PythonLinkLogic.class.getResource("PythonLinkLogic.class");
    File javaFile = new File(main.getPath());

    String absolutePath = javaFile.getAbsolutePath();
    String javaFileFolderPath = absolutePath.substring(0,absolutePath.lastIndexOf(File.separator));
    String binFolderPath = javaFileFolderPath.substring(0,javaFileFolderPath.lastIndexOf(File.separator));
    String homeFileFolderPath = binFolderPath.substring(0,binFolderPath.lastIndexOf(File.separator));

    //append the desired directories and files

    String pythonFilePath = homeFileFolderPath + "\\ModelFolder\\Main.py";
    String csvTrainPath = homeFileFolderPath + "\\ModelFolder\\H2Clean.csv";

    try{

        //construct process with necessary arguments to run Python model via command line
        ProcessBuilder pb = new ProcessBuilder("python", pythonFilePath, csvPath, csvTrainPath);
        process = pb.start();

        //construct BufferedReader to read output from Python application
        InputStream stdout = process.getInputStream();
        BufferedReader reader = new BufferedReader(new InputStreamReader(stdout,StandardCharsets.UTF_8));

        //format output to 2 decimal places
        DecimalFormat decFormat = new DecimalFormat("#.##");
        decFormat.setRoundingMode(RoundingMode.HALF_UP);

        String line = "";
        while((line = reader.readLine()) != null){

            //read in the Python output and format it
            line = line.replaceAll("[\\[\\]]", "");
            line = line.trim();
            Double dblLine = Double.parseDouble(line) * 100;
            predictionArrayList.add(decFormat.format(dblLine));
```

Figure 11 A code extract detailing the process of connecting to the Python script from the Java application

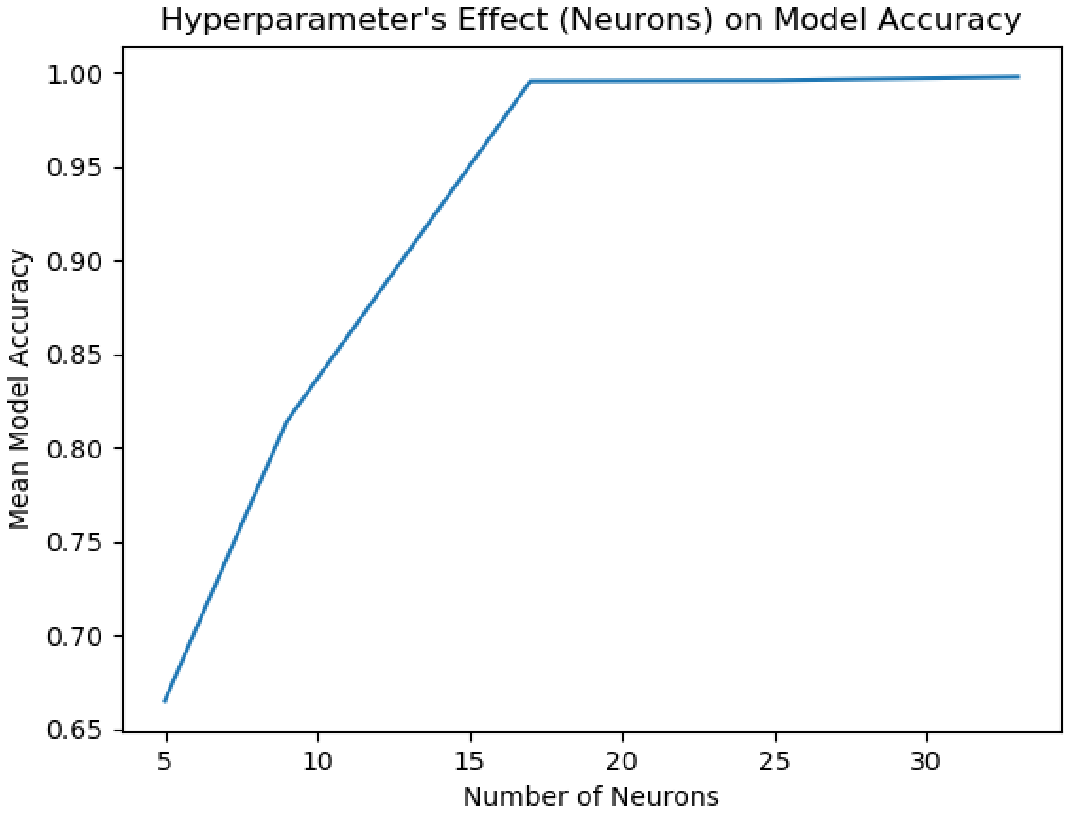


Figure 12 Neuron Hyperparameter Effect

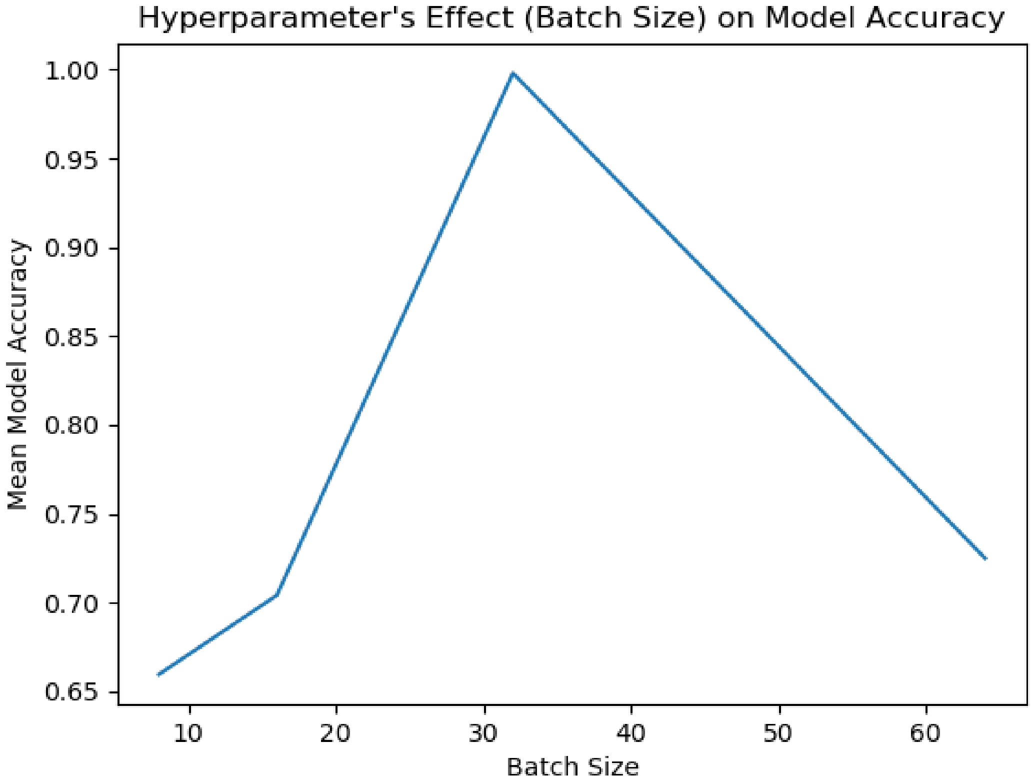


Figure 13 Batch Size Hyperparameter Effect

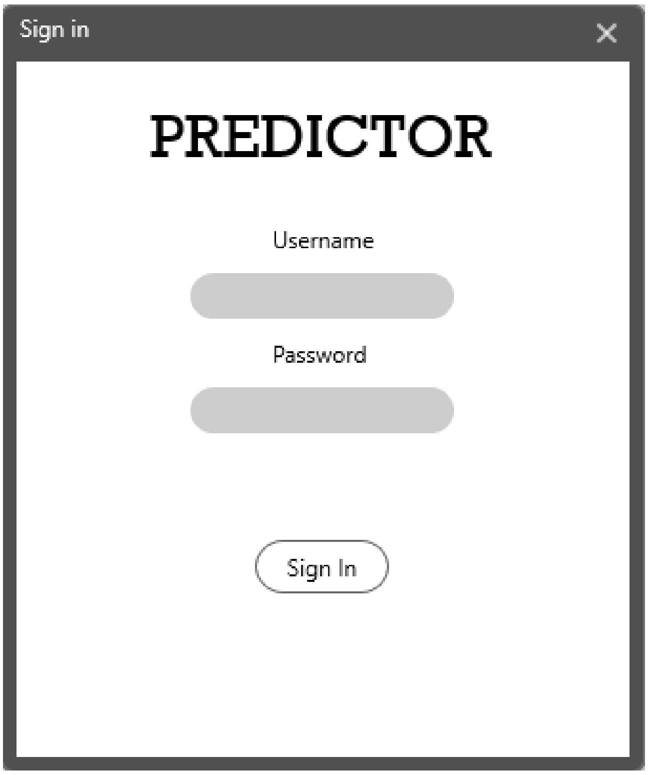
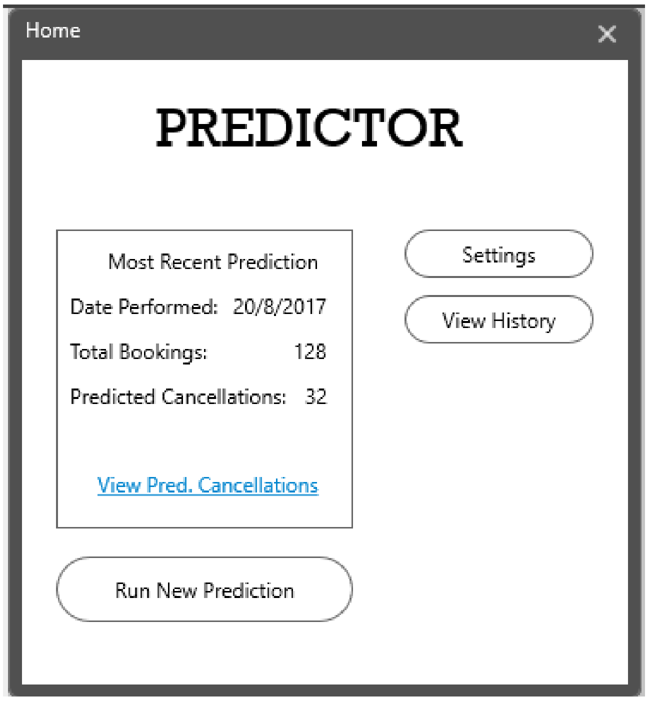


Figure 14 Wireframe of Sign In Process



Most Recent Prediction

Date Performed: 20/8/2017

Total Bookings: 128

Predicted Cancellations: 32

View Pred. Cancellations

Figure 15 Wireframe of Home Page

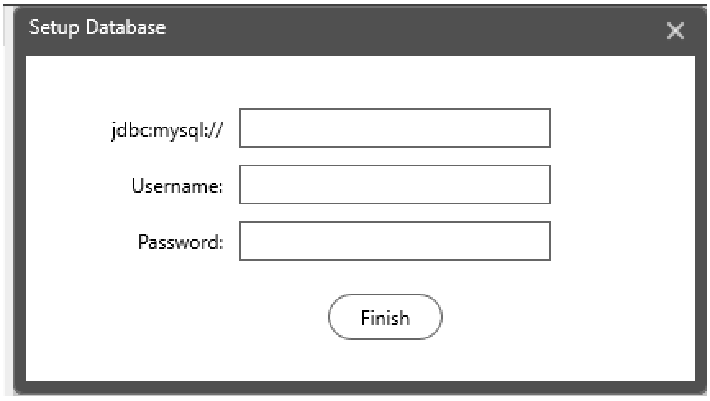
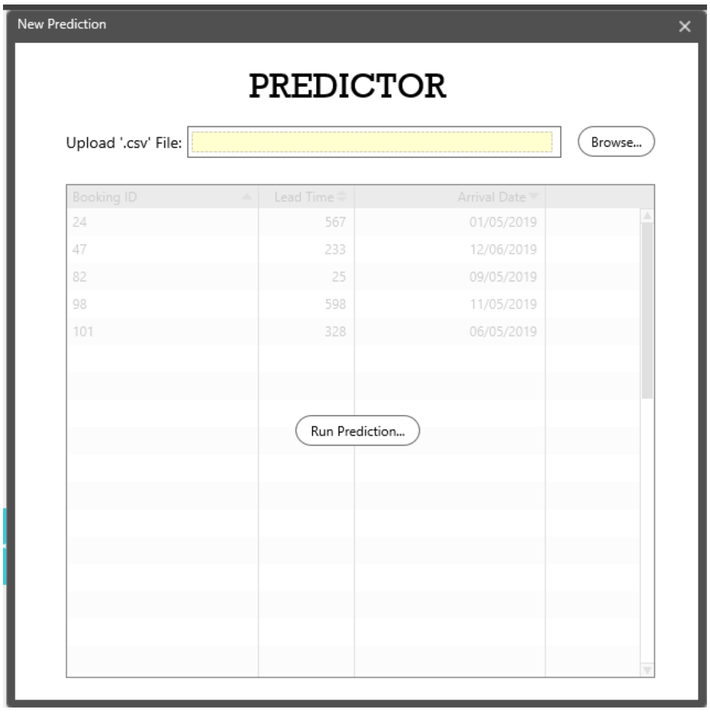


Figure 16 Wireframe of Database Setup Process



Booking ID	Lead Time	Arrival Date
24	567	01/05/2019
47	233	12/06/2019
82	25	09/05/2019
98	598	11/05/2019
101	328	06/05/2019

Figure 17 Wireframe of Prediction Page

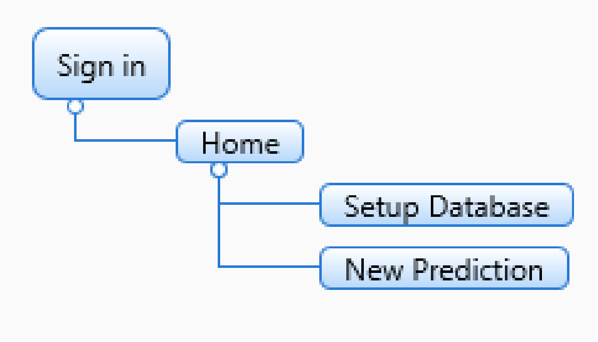
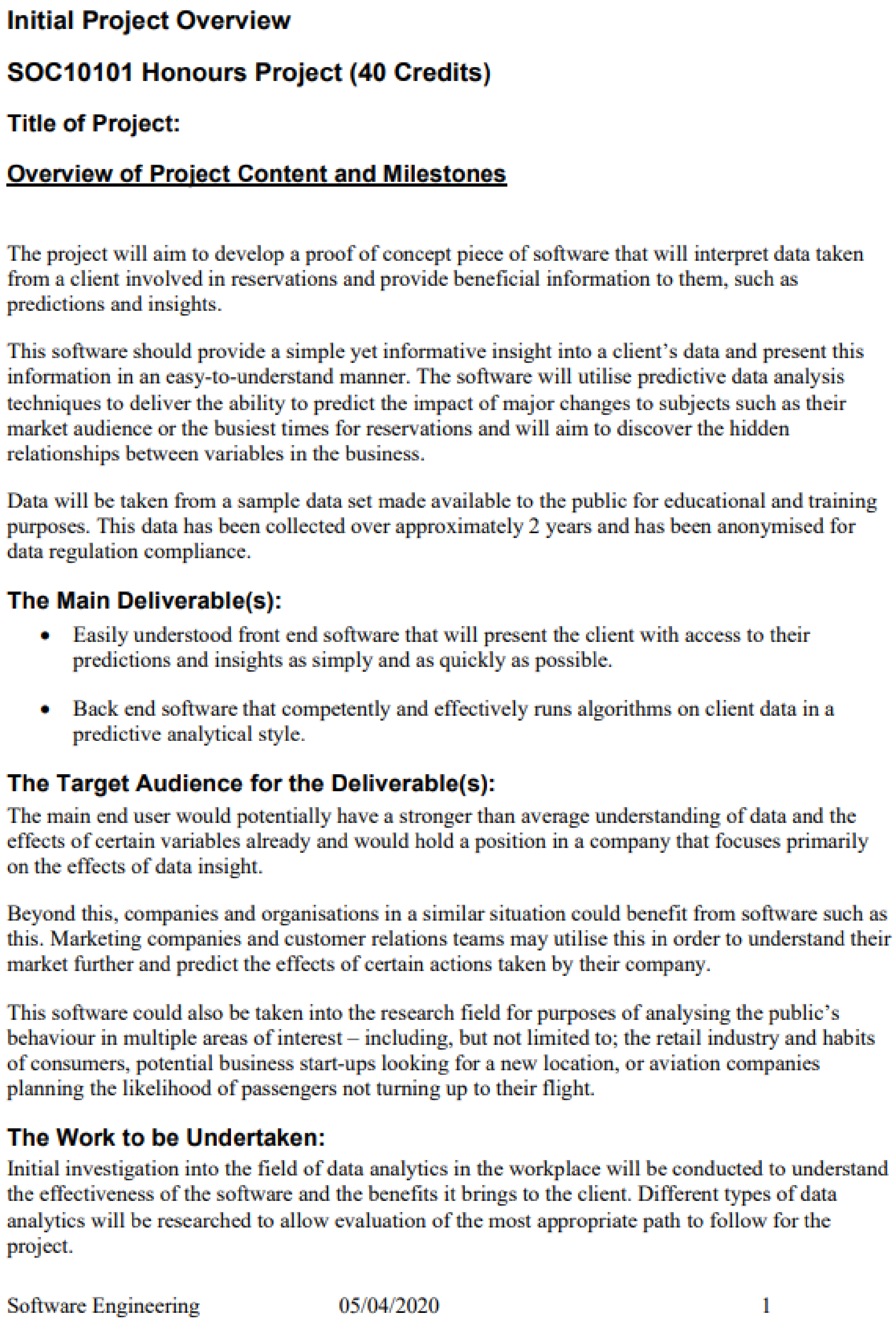


Figure 18 Hierarchical Design Tree of Software

Appendix 2 Initial Project Overview



Initial Project Overview

SOC10101 Honours Project (40 Credits)

Title of Project:

Overview of Project Content and Milestones

The project will aim to develop a proof of concept piece of software that will interpret data taken from a client involved in reservations and provide beneficial information to them, such as predictions and insights.

This software should provide a simple yet informative insight into a client's data and present this information in an easy-to-understand manner. The software will utilise predictive data analysis techniques to deliver the ability to predict the impact of major changes to subjects such as their market audience or the busiest times for reservations and will aim to discover the hidden relationships between variables in the business.

Data will be taken from a sample data set made available to the public for educational and training purposes. This data has been collected over approximately 2 years and has been anonymised for data regulation compliance.

The Main Deliverable(s):

- Easily understood front end software that will present the client with access to their predictions and insights as simply and as quickly as possible.
- Back end software that competently and effectively runs algorithms on client data in a predictive analytical style.

The Target Audience for the Deliverable(s):

The main end user would potentially have a stronger than average understanding of data and the effects of certain variables already and would hold a position in a company that focuses primarily on the effects of data insight.

Beyond this, companies and organisations in a similar situation could benefit from software such as this. Marketing companies and customer relations teams may utilise this in order to understand their market further and predict the effects of certain actions taken by their company.

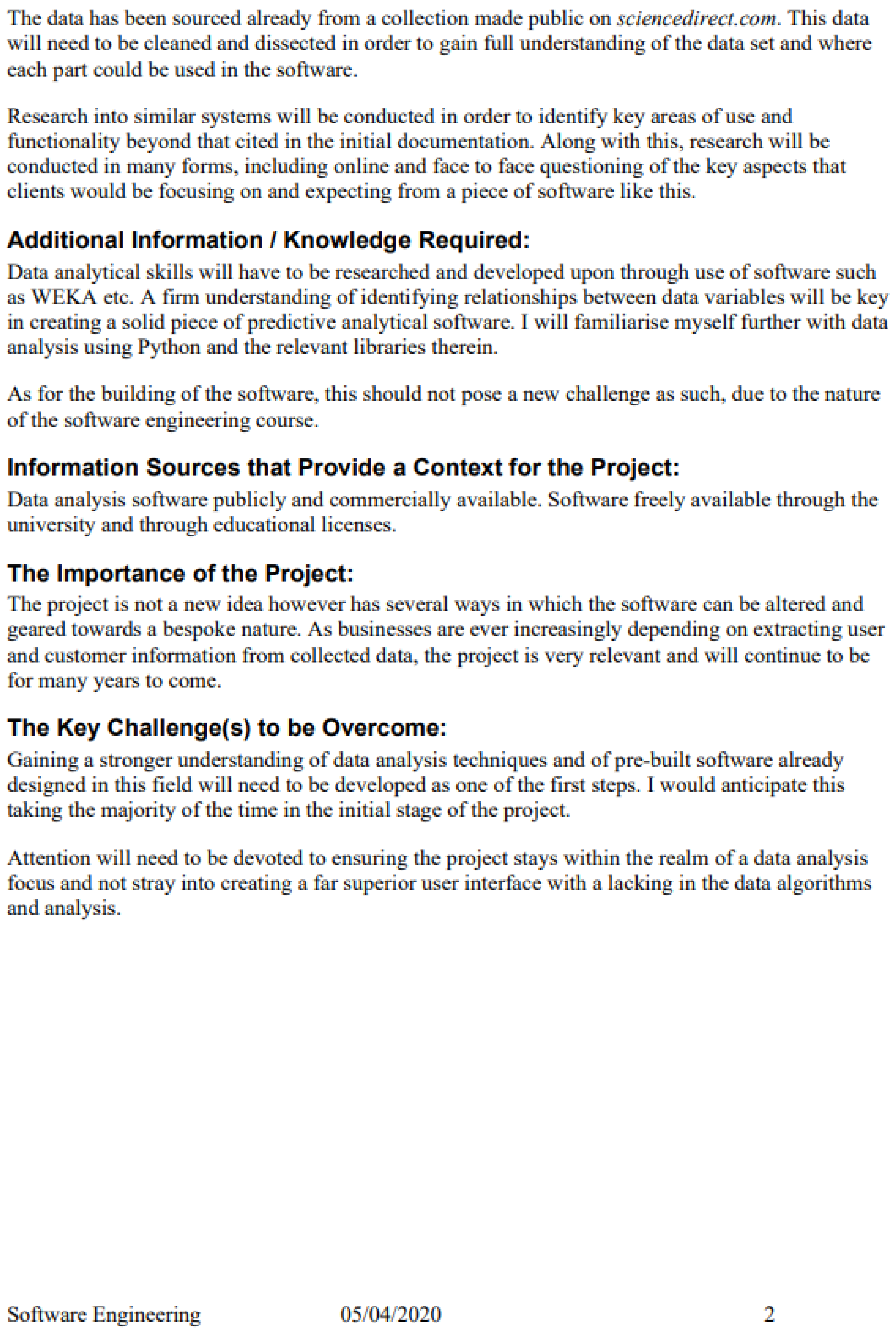
This software could also be taken into the research field for purposes of analysing the public's behaviour in multiple areas of interest – including, but not limited to; the retail industry and habits of consumers, potential business start-ups looking for a new location, or aviation companies planning the likelihood of passengers not turning up to their flight.

The Work to be Undertaken:

Initial investigation into the field of data analytics in the workplace will be conducted to understand the effectiveness of the software and the benefits it brings to the client. Different types of data analytics will be researched to allow evaluation of the most appropriate path to follow for the project.

Software Engineering 05/04/2020 1

Figure 19 Initial Project Overview Page 1



The data has been sourced already from a collection made public on *sciencedirect.com*. This data will need to be cleaned and dissected in order to gain full understanding of the data set and where each part could be used in the software.

Research into similar systems will be conducted in order to identify key areas of use and functionality beyond that cited in the initial documentation. Along with this, research will be conducted in many forms, including online and face to face questioning of the key aspects that clients would be focusing on and expecting from a piece of software like this.

Additional Information / Knowledge Required:

Data analytical skills will have to be researched and developed upon through use of software such as WEKA etc. A firm understanding of identifying relationships between data variables will be key in creating a solid piece of predictive analytical software. I will familiarise myself further with data analysis using Python and the relevant libraries therein.

As for the building of the software, this should not pose a new challenge as such, due to the nature of the software engineering course.

Information Sources that Provide a Context for the Project:

Data analysis software publicly and commercially available. Software freely available through the university and through educational licenses.

The Importance of the Project:

The project is not a new idea however has several ways in which the software can be altered and geared towards a bespoke nature. As businesses are ever increasingly depending on extracting user and customer information from collected data, the project is very relevant and will continue to be for many years to come.

The Key Challenge(s) to be Overcome:

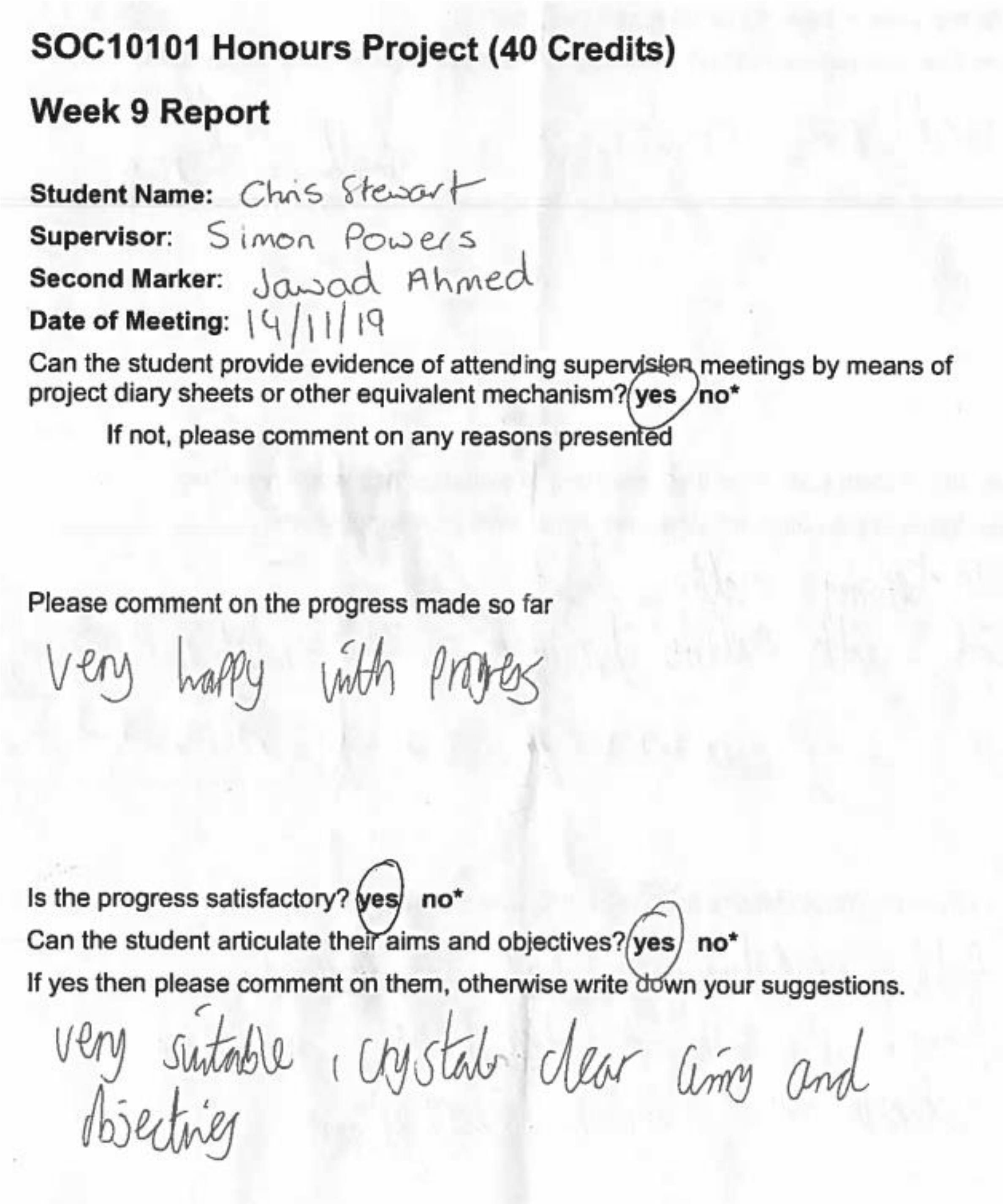
Gaining a stronger understanding of data analysis techniques and of pre-built software already designed in this field will need to be developed as one of the first steps. I would anticipate this taking the majority of the time in the initial stage of the project.

Attention will need to be devoted to ensuring the project stays within the realm of a data analysis focus and not stray into creating a far superior user interface with a lacking in the data algorithms and analysis.

Software Engineering 05/04/2020 2

Figure 20 Initial Project Overview Page 2

Appendix 3 Second Formal Review Output



SOC10101 Honours Project (40 Credits)

Week 9 Report

Student Name: Chris Stewart

Supervisor: Simon Powers

Second Marker: Jawad Ahmed

Date of Meeting: 14/11/19

Can the student provide evidence of attending supervision meetings by means of project diary sheets or other equivalent mechanism? [✓ yes] / no*

If not, please comment on any reasons presented

Please comment on the progress made so far

Very happy with progress

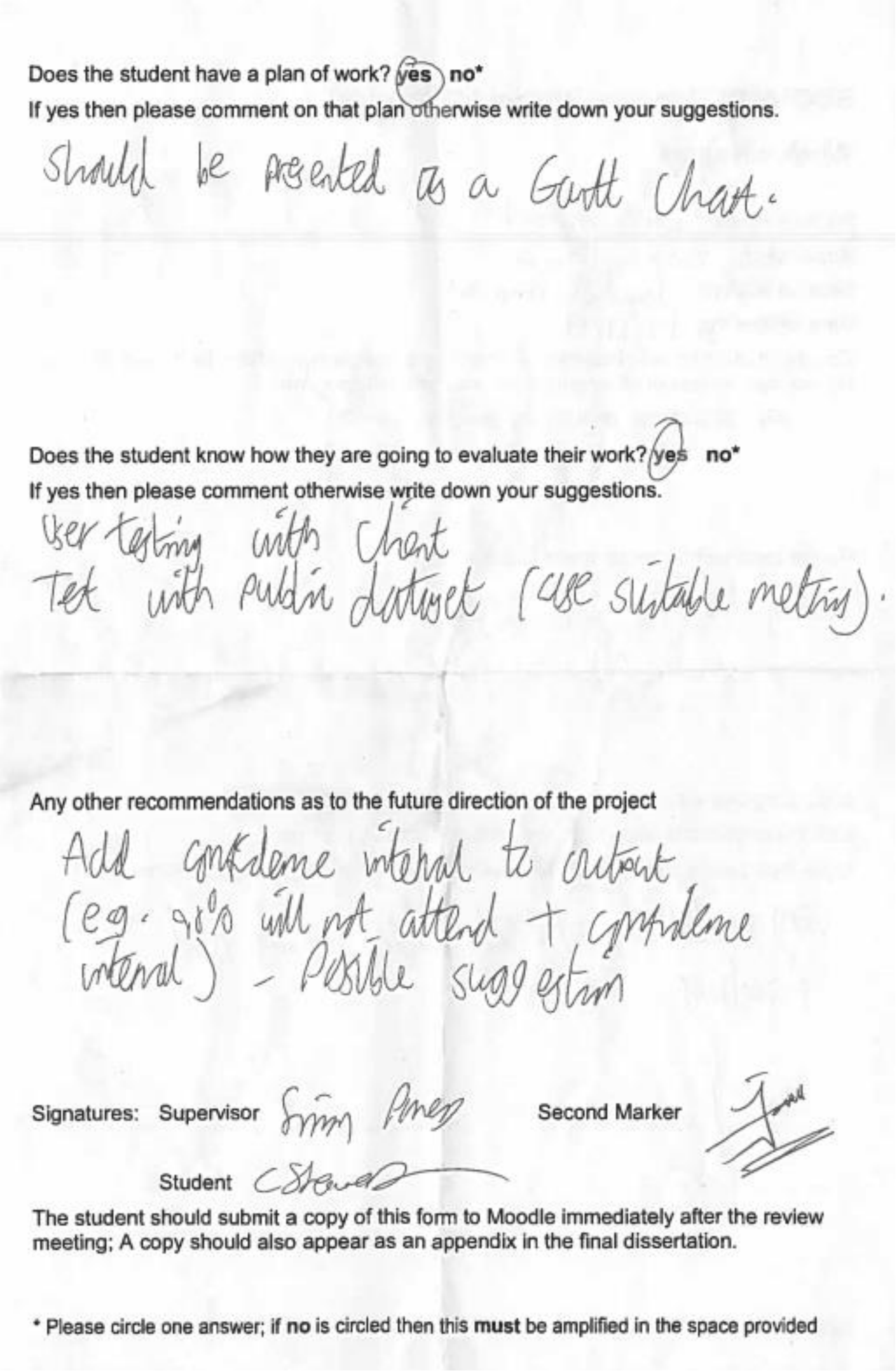
Is the progress satisfactory? [✓ yes] no*

Can the student articulate their aims and objectives? [✓ yes] no*

If yes then please comment on them, otherwise write down your suggestions.

very suitable , crystal clear aims and objectives

Figure 21 Second Formal Review Page 1



Does the student have a plan of work? [✓ yes] no*

If yes then please comment on that plan otherwise write down your suggestions.

Should be presented as a Gantt Chart.

Does the student know how they are going to evaluate their work? [✓ yes] no*

If yes then please comment otherwise write down your suggestions.

User testing with Client

Test with public dataset (use suitable metrics).

Any other recommendations as to the future direction of the project

Add confidence interval to output (e.g. 98% will not attend + confidence interval) - Possible suggestion

Signatures: Supervisor Simon Jones Second Marker

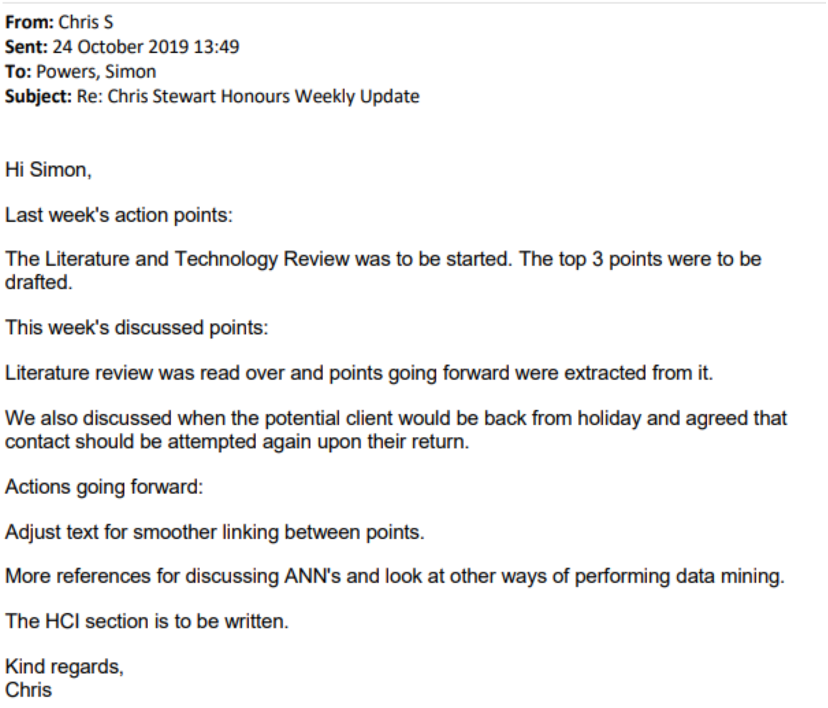
Student C Stewart

The student should submit a copy of this form to Moodle immediately after the review meeting; A copy should also appear as an appendix in the final dissertation.

* Please circle one answer; if **no** is circled then this **must** be amplified in the space provided

Figure 22 Second Formal Review Page 2

Appendix 4 Project Management Evidence



From: Chris S
Sent: 24 October 2019 13:49
To: Powers, Simon
Subject: Re: Chris Stewart Honours Weekly Update

Hi Simon,

Last week's action points:

The Literature and Technology Review was to be started. The top 3 points were to be drafted.

This week's discussed points:

Literature review was read over and points going forward were extracted from it.

We also discussed when the potential client would be back from holiday and agreed that contact should be attempted again upon their return.

Actions going forward:

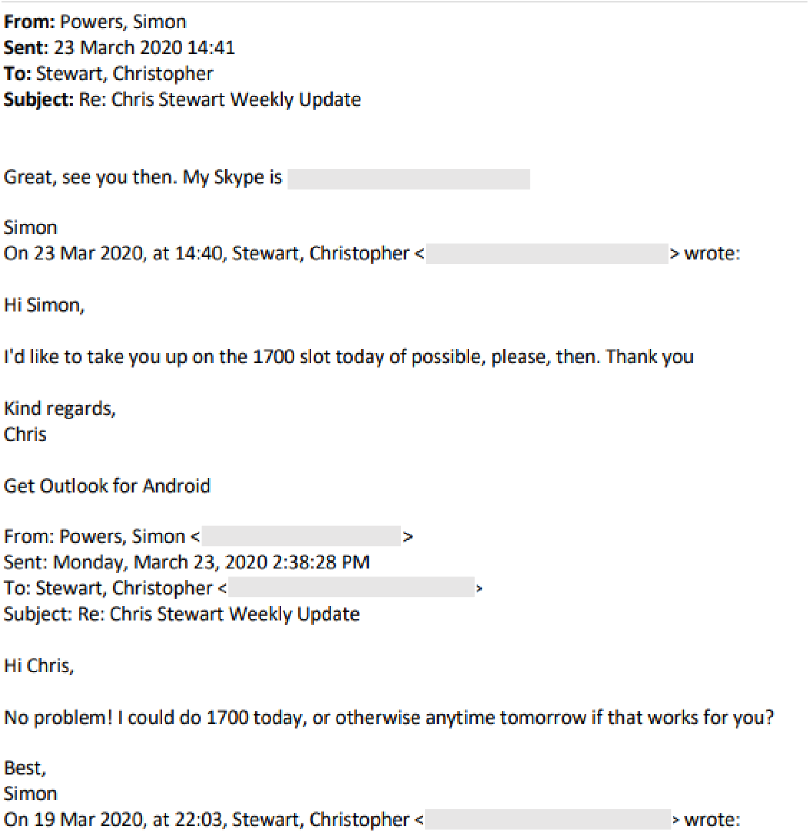
Adjust text for smoother linking between points.

More references for discussing ANN's and look at other ways of performing data mining.

The HCI section is to be written.

Kind regards,
Chris

Figure 23 Weekly Update Email to Supervisor



From: Powers, Simon
Sent: 23 March 2020 14:41
To: Stewart, Christopher
Subject: Re: Chris Stewart Weekly Update

Great, see you then. My Skype is [redacted]

Simon
On 23 Mar 2020, at 14:40, Stewart, Christopher <[redacted]> wrote:

Hi Simon,

I'd like to take you up on the 1700 slot today of possible, please, then. Thank you

Kind regards,
Chris

Get Outlook for Android

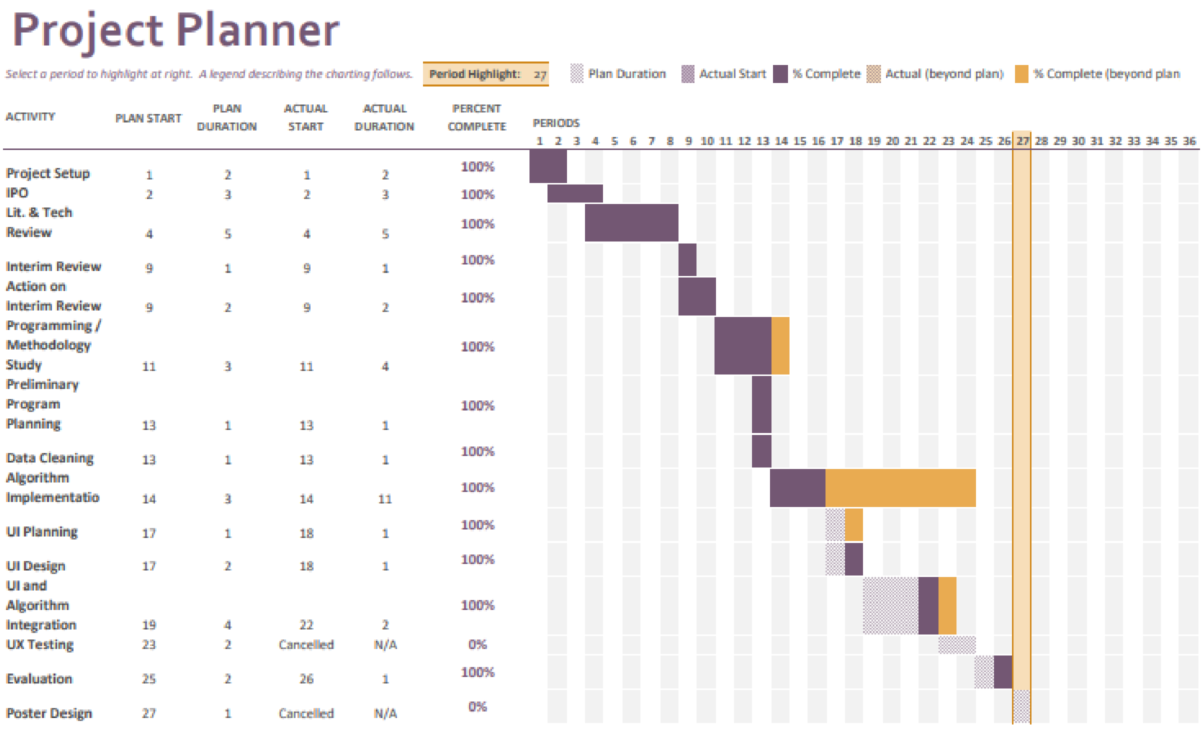
From: Powers, Simon <[redacted]>
Sent: Monday, March 23, 2020 2:38:28 PM
To: Stewart, Christopher <[redacted]>
Subject: Re: Chris Stewart Weekly Update

Hi Chris,

No problem! I could do 1700 today, or otherwise anytime tomorrow if that works for you?

Best,
Simon
On 19 Mar 2020, at 22:03, Stewart, Christopher <[redacted]> wrote:

Figure 24 Email Rescheduling Video Meeting



Project Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 27

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Project Setup	1	2	1	2	100%
IPO	2	3	2	3	100%
Lit. & Tech Review	4	5	4	5	100%
Interim Review	9	1	9	1	100%
Action on Interim Review	9	2	9	2	100%
Programming / Methodology Study	11	3	11	4	100%
Preliminary Program Planning	13	1	13	1	100%
Data Cleaning	13	1	13	1	100%
Algorithm Implementatio	14	3	14	11	100%
UI Planning	17	1	18	1	100%
UI Design	17	2	18	1	100%
UI and Algorithm Integration	19	4	22	2	100%
UX Testing	23	2	Cancelled	N/A	0%
Evaluation	25	2	26	1	100%
Poster Design	27	1	Cancelled	N/A	0%

Figure 25 Gantt Chart