**H48W 35 Computing: Software Development  
Graded Unit 2**

**Project Stage: Inception Planning**

**Action Plan Report**

**Student name: Daria Vekic**

**Student number: 586661**

**Date: January / February 2023**

Table of Contents

[Overview 1](#_Toc126849128)

[Introduction 1](#_Toc126849129)

[Initial Project Brief 2](#_Toc126849130)

[Section 1 – Assignment Brief Analysis 3](#_Toc126849131)

[Development Routes 3](#_Toc126849132)

[Option 1 – Web Solution with Database 3](#_Toc126849133)

[Option 2 – Java Project with Database 6](#_Toc126849134)

[Option 3 – Java Project with Data Structures 9](#_Toc126849135)

[Summary 9](#_Toc126849136)

[Software Development Lifecycle Model (SDLM) 12](#_Toc126849137)

[Waterfall Methodology 12](#_Toc126849138)

[Unified Process 12](#_Toc126849139)

[Key Deliverables 14](#_Toc126849140)

[Summary 15](#_Toc126849141)

[Natural Language Analysis 17](#_Toc126849142)

[Potential Entities 17](#_Toc126849143)

[Potential Actions 19](#_Toc126849144)

[Section 2 – Initial Planning Models 21](#_Toc126849145)

[Initial Functional Requirements 21](#_Toc126849146)

[Initial Non-Functional Requirements 21](#_Toc126849147)

[Initial Top Level Use Case Diagram 22](#_Toc126849148)

[Section 3 – Information Gathering 24](#_Toc126849149)

[Initial Client Interview 24](#_Toc126849150)

[Client Questions 25](#_Toc126849151)

[Meeting Number 1 Agenda 26](#_Toc126849152)

[Meeting Number 1 Minutes 28](#_Toc126849153)

[Summary of Client Interaction 30](#_Toc126849154)

[Background Research 31](#_Toc126849155)

[General Job Responsibilities 31](#_Toc126849156)

[Immigration Solicitors 31](#_Toc126849157)

[Criminal Solicitors 32](#_Toc126849158)

[Personal Injury Solicitors 32](#_Toc126849159)

[Forms of Payment 32](#_Toc126849160)

[Investigation of Similar Systems 34](#_Toc126849161)

[iManage 34](#_Toc126849162)

[Ease of Use 34](#_Toc126849163)

[Accessibility 34](#_Toc126849164)

[Functionality 35](#_Toc126849165)

[LawWare 36](#_Toc126849166)

[Fee Recording 36](#_Toc126849167)

[Search Functionality 36](#_Toc126849168)

[Summary 37](#_Toc126849169)

[Meeting Number 2 Agenda 38](#_Toc126849170)

[Meeting Number 2 Minutes 44](#_Toc126849171)

[Section 4 – Project Aims 48](#_Toc126849172)

[Refined Functional Requirements 48](#_Toc126849173)

[Prioritised List of Functional Requirements 49](#_Toc126849174)

[Final Functional Requirements 51](#_Toc126849175)

[Final Non‑Functional Requirements 52](#_Toc126849176)

[Refined Use Case Diagram 55](#_Toc126849177)

[Conceptual Model 56](#_Toc126849178)

[System Proposal 58](#_Toc126849179)

[Section 5 – Resources 60](#_Toc126849180)

[People 60](#_Toc126849181)

[Hardware 61](#_Toc126849182)

[Software 62](#_Toc126849183)

[Informational 63](#_Toc126849184)

[Other 67](#_Toc126849185)

[Section 6 – Feasibility Analysis 68](#_Toc126849186)

[Technical Feasibility 68](#_Toc126849187)

[Financial Feasibility 69](#_Toc126849188)

[Time Feasibility 69](#_Toc126849189)

[Legal Feasibility 70](#_Toc126849190)

[Summary 71](#_Toc126849191)

[Bibliography 72](#_Toc126849192)

**Table of Figures**

[Figure 1 Binary files link with libraries and become platform‑dependant executable 6](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849193)

[Figure 2 Executables that can execute on only one platform 7](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849194)

[Figure 3 Java compiler creates platform‑independent Java bytecode 7](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849195)

[Figure 4 Java bytecode file is executable if JRE exists 8](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849196)

[Figure 5 The UP is iterative and incremental by nature 13](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849197)

[Figure 6 Top level use case diagram 22](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849198)

[Figure 7 iManage GUI 35](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849199)

[Figure 8 Refined use case diagram 55](#_Toc126849200)

[Figure 9 Top-level conceptual model identifies important entities 57](file:///E:\Graded%20Unit\1.%20Planning%20Phase\1a%20Inception%20Planning\586661\586661%20-%20Planning%20Phase%20Part%20A%20-%20Action%20Plan%20Report.docx#_Toc126849201)

# Overview

This document relates to the Graded Unit 2 project as part of the HND Computing: Software Development H48W 35. It aims to satisfy Stage 1, Part A of the overall project by providing a detailed analysis of the project assignment brief and should be read in conjunction with the Project Plan Report submitted separately.

# Introduction

The contents of this document are intended to be used as an Action Plan for the Graded Unit 2 project. It documents a variety of fact‑finding tasks. The project assignment brief that this project is based on is included in this report.

Section 1 of this report aims to analyse the project assignment brief by providing an **interpretation of the brief**. In doing so, various development routes have been identified with a detailed evaluation of the chosen development route provided. Section 1 also discusses the Software Development Lifecycle Model (SDLM) and specifies the approach that will be adopted in this project, providing details of the key deliverables that will be produced throughout the overall process. It also includes a Natural Language Analysis and Summary which not only identifies potential entities that are of significance to the client, but also guides our analysis of the brief to pick out the most important requirements of the system to be developed.

Section 2 of this report addresses the creation of the **initial planning models**. This section includes an initial functional and non‑functional requirements list, and an initial top‑level Use Case diagram.

Section 3 includes details on the execution of **information gathering**. This includes the client interaction that has taken place. The relevant meeting agendas, meeting minutes, and question list issued to the client can be found in full in this section. It further sets out the background research that has been undertaken and the results of an investigation of similar systems. Section 3 culminates in an analysis of the information gathered, providing a justification of initial decisions and highlights new information that has been established through information gathering techniques.

Section 4 of the report sets out what we aim to achieve in terms of the **functionality in the final solution**. As such, there is a list of refined, though not final, functional requirements and a refined use case diagram. There is a system proposal contained in Section 4 which takes into consideration all of the information obtained throughout the Planning phase so far and lays out the **aims of the project**. This system proposal reflects the refined functional and non‑functional requirements along with the refined use case diagram.

Section 5 follows on to identify **resources** – both physical and informational resources – that will be required to complete this project. The resources identified in this section will be the only resources that can be used within the scope of this Graded Unit project.

The Project Plan should be read in conjunction with this report. This identifies: the schedules for each stage and overall completion, the milestones and deliverables to show a milestone has been reached, the main tasks along with relevant subtasks, the required resources to successfully complete the project, and a Gantt Chart. The Project Plan reflects the chosen software development methodology that is discussed in Section 1 of this report.

# Initial Project Brief

McRae & Dick Solicitors are a small independent law firm based in Forth Valley who would like a case management system that allows their employees to control their workload more efficiently. They have five Solicitors: two specialise in Criminal law, two specialise in Immigration law, and one specialises in Personal Injury law. The firm also employs two Legal Secretaries who will need access to the system. Their Solicitors can only take on cases relevant to their specialism.

Regardless of a Solicitor's area of expertise, all employees should be able to create a new case, view a specific case, update the details of a case, and delete a case. There should also be the ability to generate and view the following information:

* number of total open cases;
* number of cases opened in a specified time period;
* number of cases closed in a specified time period; and
* number of cases allocated to a specific Solicitor.

Each case has a unique reference number, a Solicitor responsible for it, and the personal details of the client. When an employee creates a new case, they would like the system to assign the unique reference number to it.

The firm uses modestly specified equipment running Windows 11. They have specified that colour is not important at this stage of the project but their company colours are black, white, and green.

# Section 1 – Assignment Brief Analysis

This section provides an analysis of the Initial Project Brief. It begins by offering a discussion on three potential development routes that could be adopted in this project: a web solution with a backend database, a Java program with a backend database, and a Java program that implements data structures. The chosen development route is then outlined and justified. Similarly, a discussion of two potential methodologies is then set out with a justification for the chosen methodology included. Section 1 concludes with a Natural Language Analysis, which leads the document smoothly into Section 2 – Initial Planning Models, and acts as the starting point in devising the initial requirements of the new system.

## Development Routes

Having ascertained from the Initial Project Brief that the client wishes for a case management system to be developed, below is a discussion of three potential development routes that could be used in a project of this nature.

### Option 1 – Web Solution with Database

There is potential to develop the client’s system as a web-based solution with a back‑end database. This development route would be beneficial as it would enable any functionality relating to creating, retrieving, updating, and deleting data (CRUD). These four basic operations are obviously very important in this context of a case management system where a large amount of data is being handled.

A server‑side language such as PHP – Hypertext Pre‑Processor – could be used as it is a general-purpose scripting language where a script will run in response to an event. It is able to generate dynamic web page content and supports the aforementioned operations (CRUD) on the server. It also allows the web page to: collect form data, send/receive cookies, add/delete/modify data in the database, and to encrypt data. This functionality is crucial to the success of the client’s new system as there is a need to add, modify, delete legal cases. PHP enables us to output a range of multimedia information such as images, PDFs, and videos. The language interleaves HTML markup and PHP instructions.

Another common server-side language is Structured Query Language (SQL). SQL is the standard way to talk to a database and it can be embedded in PHP documents or alternatively can be written in .sql documents. If this development route were to be chosen, SQL would be necessary in order to get the desired data from the database. SQL has the ability to execute queries against a database; retrieve data from a database; insert records in a database; update records in a database; delete records from a database, amongst other functionalities.

Using a relational database management system (RDMS) should be carefully considered for the client’s solution as it provides all of the facilities to meet the needs of the client. Using a centralised and structured set of data is very time and cost effective as employees would not only be able to quickly and easily retrieve data, but data entry would be immediate, so the possibility of creating a backlog of paperwork is minimised. RDMSs are generally organised into tables with any number of rows and columns, so if the client’s firm were to expand, the RDMS would easily have the capacity to grow as needed. Additionally, abiding by data‑integrity rules (constraints) will define the relationally correct state for the database, meaning that users are only able to perform the operations which leave the database in its correct and consistent state. For example, the constraint known as entity integrity means that a primary key must be unique and no part of it can be left null. In the context of a case management system, an employee would not be able to create a new case without a case reference number. In other words, a case cannot exist without its case reference number, which reflects the business rules of the client.

This development route would also require the use of client‑side programming languages. This is the language that builds the program which runs on the client machine and deals with the user interface.

One useful client‑side language is Hypertext Markup Language (HTML) which essentially communicates to the user’s computer what things are. HTML provides developers with access to a large amount of functionality that is already built directly into the browser. HTML is the standard markup language for creating web pages and describes the structure of the page. It consists of a series of elements defined by a start and end tag (for example, a paragraph is marked <p>This is a paragraph</p>). Furthermore, semantic HTML communicates to a screen reader and a browser how to navigate the page. This ensures accessibility for users who navigate a site with a keyboard only.

Although this option has valid advantages, it comes with risks that should be considered carefully. Patrick Engebretson explains very clearly that the web comes with many dangers, referring to the web as “one of the most common attack vectors available today because *everything* is connected to the Internet” (Engebretson, 2013). As time goes on, what would previously have been installed locally can now be done online (for example, instead of using Microsoft Word, many users opt to use Google Docs online). Engebretson highlights that many companies are now “leveraging the power of an executable web”, but simultaneously failing to recognise that as we “push everything to the web and systems are mashed up and deployed with worldwide accessibility, new attacks are developed and distributed at a furious pace” (Engebretson, 2013).

One of the most common web vulnerabilities is a SQL injection – an attack technique often used by unethical hackers to exploit a weakness in code used to produce the website. A plain SQL injection will display an error message relating to the SQL syntax. The web page will then accept data from a client and then execute SQL queries without validating that input. The hacker is then able to extract, add, modify, or delete content for the database. Similarly, a blind SQL injection does not provide any error message, but the attacker is presented with a generic page that is specified by the developer. Any SQL injection poses extreme threat to the firm and could result in: an authentication bypass that allows the attacker to log in to the system with no valid credentials; information disclosure where an attacker obtains sensitive and confidential information contained in a database; compromised data integrity where the database contents are altered; and/or remote command execution where an attacker can perform command execution via a database to compromise the host operating system.

With this in mind, the client may even have to consider employing a Database Administrator for the maintenance of the RDMS.

### Option 2 – Java Project with Database

Another potential development route that also makes use of a database is a Java project.

In addition to the benefits of adopting a RDMS explained in Option 1, a significant advantage of developing the solution using the strongly typed Java language is that it is platform‑independent. If using other programming languages, running a case management system would generally require a lot of modifications to run if the client ever wished to use the system on a different type of computing platform. This is because when programming in other languages, specific code is required depending on the underlying platform. So, if the new system were to be built using C or C++, there would be a requirement to compile and link the programs which results in an executable program that is unique to that platform. The compiler converts the program into machine code, which is the code understood by the CPU. The resulting binary files may be combined with many other files (like libraries of prewritten code) with a linker, creating the platform‑dependent program known as an executable (see Figure 1). This is what the end user can execute. In other words, a platform‑dependant executable can be executed on only one platform (see Figure 2).

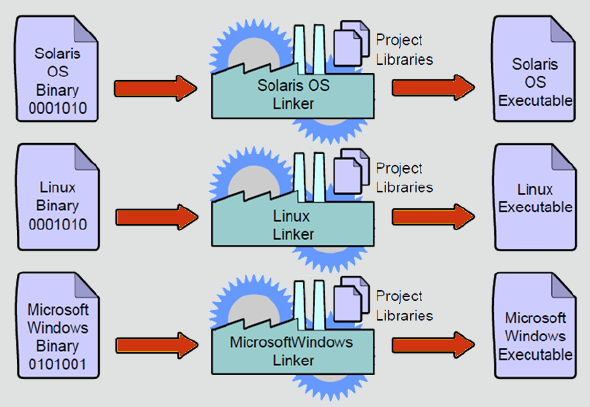


Figure Binary files link with libraries and become platform‑dependant executable

Figure taken from: Oracle Academy Java Foundations course notes, Section 1-3, slide 6

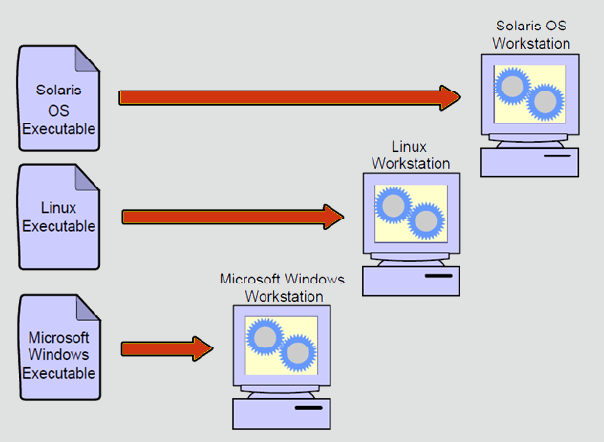


Figure Executables that can execute on only one platform

Figure taken from: Oracle Academy Java Foundations course notes, Section 1-3, slide 7

A significant advantage of using Java is that it has the ability to run on a variety of different CPU and operating system combinations with very little – if any – modifications needed. This is because a Java program is compiled with a Java compiler, meaning the result is platform‑independent Java bytecode and not machine code (Figure 3). Java bytecode is then interpreted by the Java Virtual Machine which can then execute that Java program on any platform. This is how Java programs have become known as portable – they are executable on any platform. All that is required is the Java Virtual Machine on the platform where the system will run along with the Java class libraries, known together as the Java Runtime Environment (JRE).

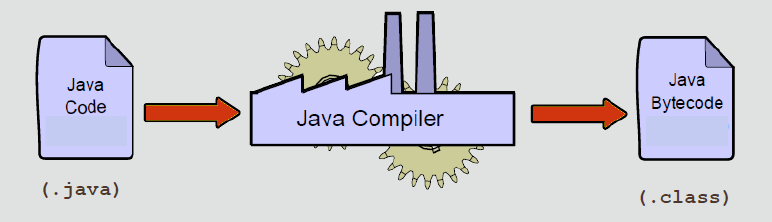


Figure Java compiler creates platform‑independent Java bytecode

Figure taken from: Oracle Academy Java Foundations course notes, Section 1-3, slide 8

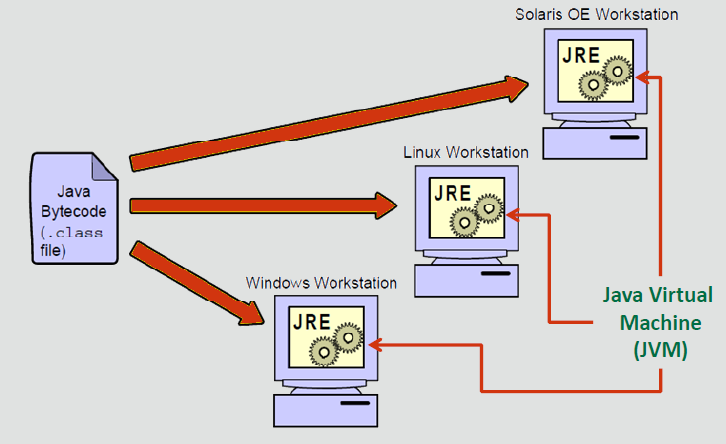


Figure Java bytecode file is executable if JRE exists

*Figure taken from: Oracle Academy Java Foundations course notes, Section 1-3, slide 9*

This is advantageous to the client as if they change their office equipment and begin to use, for example, macOS, the firm would still have the ability to run this software. Moving computer systems would not create significant difficulty. Further, if they ever wanted to turn this into a mobile application to allow solicitors to enter case details/updates whilst out of office (for example their Criminal solicitors visiting clients in prison, or any Solicitor who is at Court), there is a higher chance of this being feasible if the initial system is written in Java to begin with.

However, Java’s overall performance is slower than languages such as C and C++ because it has to be interpreted during runtime. Further, as a result of running on top of the Java Virtual Machine, it consumes more memory and processing time, meaning there can be a need to have better quality hardware.

It cannot be denied though that Java is an efficient language to use to build this system because it is object‑oriented. This means that developers will use modular code that can be reused, which ultimately saves time on the project as it minimises the amount of testing (if a module achieves its purpose, it does not have to be retested repeatedly).

Overall it is a robust and secure language that allows developers to have high degree of control over data. This is vital for data security and data integrity.

### Option 3 – Java Project with Data Structures

Another possible development route for this project is a Java project using data structures rather than a database. This route would benefit from the aforementioned advantages of the Java programming language along with the advantages of data structures.

The use of data structures provides the developer with a high degree of control over the client’s data. In comparison to a database management system, data structures do not pose as high a risk of SQL injections to the client. Yet, all of the same functionality available through a database can still be achieved through data structures: data can still be stored and fetched in an efficient manner if the most appropriate data structure is implemented.

As data structures are very common in programming, Java has pre‑built classes that developers can make use of. This will save the developer the time and effort of having to create storage structures. In other words, there is commonly already a data structure that exists which performs the task a developer has to implement; the drawback here is that the efficiency and success of the system depends on the developer choosing to use the correct data structure for the correct task. For example, the Collection interface in the java.util package can be used to define a group of objects, including sets and lists. This interface defines three main collections: Set, List, and Queue. As it is an interface, it forces any class that implements it to define methods such as add(). In fact, it declares 15 methods that must be implemented by any class that implements the Collection interface. We can then, for example, use the concrete class ArrayList which implements the List interface. A List provides important qualities that would be relevant in a case management system as they can: grow and shrink dynamically, maintain a specific order, and allow duplicate elements. We do not need to know the size of the ArrayList when initialised, therefore it is an appropriate data structure to use if the legal firm cannot predict how many new clients they will gain in the near future. The only information needed to initialise an ArrayList is the object type that it stores. In addition, a HashSet – a concrete class that implements the Set interface – would provide a quick way of finding an element than an ArrayList, especially if the list is very large. Educator and Software Engineer Jay Wengrow highlights that depending on “how you choose to organize your data, your program may run faster or slower by orders of magnitude” (Wengrow, 2020). When the most appropriate data structure is implemented, the code performs efficiently and we have the control to organise the client’s data specifically to meet the client’s needs in a more secure fashion compared to implementing a back‑end database.

### Summary

Although it consumes more memory and has a higher processing time, the benefits of using a platform‑independent and object‑oriented language far outweigh the benefits of using a web‑based solution or opting for another programming language like C++. The client has specified they use modest equipment running on Windows 11 so memory management and slower processing time should not hinder the user experience. Java is the global standard for developing and delivering a variety of applications, whereas a solution using a database exposes the client to dangerous risks like SQL injection attacks that could ultimately destroy the business, especially given the nature of data that a legal firm deals with.

Due to this risk of attack, **I intend to develop the client’s solution through a Java project with data structures** which can achieve the same functionality as a database (option 3). This is the most appropriate route for all project stakeholders (including project team members and the client) as Java is a robust language and combined with the use of data structures which allow developers a high degree of control over data will ensure the firm is far more cyber resilient.

The table below shows at a glance the key benefits and drawbacks of each development route discussed above.

|  |  |  |
| --- | --- | --- |
| **Option** | **Benefits** | **Drawbacks** |
| 1 – Web with database | * Supports all CRUD functions * Capacity to grow * Centralised and structured set of data * Time and cost effective * Quickly and easily retrieve data * Data entry would be immediate * Data integrity rules ensure correct and consistent state * HTML good for accessibility | * High risk of SQL injection attach (plain or blind) * Serious risks to client including data breach * Attacker could freely access all data * Possibility of catastrophic consequences on business * Requires high level of maintenance |
| 2 – Java with database | * Strongly typed language – more control over data * Platform independent * Object‑oriented * Modular code can be reused * Robust and secure language | * Java consumes more memory * Takes longer to process * Risk of SQL injection remains |
| 3 – Java with data structures | * High degree of control over data * Risk of SQL injection eliminated * Pre‑built classes to use * Supports all CRUD functions * Capacity to grow as needed * Quickly and easily retrieve data | * Java consumes more memory * Takes longer to process * Success relies on developer choosing most appropriate data structure |

## Software Development Lifecycle Model (SDLM)

Potential SDLMs that could be followed in this project are discussed below and a justification of the chosen SDLM is then provided.

The SDLM, also known as the Systems Development Lifecycle (SDLC), is important to understand as it provides the project team with an overarching process to follow at all times. It involves the understanding of how the information system will support the business needs. The SDLC involves a System Analyst who analyses the business situation, identifies the opportunities for improvement, and then designs an information system to implement the improvements. Regardless of what specific methodology is adopted, there are generally four key phases: Planning, Analysis, Design, and Implementation.

### Waterfall Methodology

The waterfall methodology is a highly structured design methodology. It follows a sequential process where one phase cannot begin before the preceding phase is complete. As a result, the requirements of the new system are established well in advance of programming commencing. This results in every project team member having an understanding of what must be completed and the deadline by which it should be completed, allowing for effective planning of time and an accurate estimation of costs. Its highly structured nature minimises any changes made to the requirements as the project progresses.

However, this methodology is time consuming as the design has to be complete in its entirety before development can begin. This can mean the new software may not be ready for many months or years. During this time, the business needs and business environment may have changed since the initial phases, so work carried out early in the project may be irrelevant. Additionally, if a requirement is missed, it becomes very costly for the project team to revisit this; any change – expected or unexpected – creates a significant amount of work. There is also not a large amount of client or end user interaction, so once the end goal is established at the start of the project, the client is not as involved in the software development process. It then becomes difficult to know if the software being developed is indeed going to satisfy the client's needs. The waterfall methodology is more suitable for organisations such as construction companies: if the goal is, for example, to build houses then there are distinct milestones and many tasks are dependant on the previous task being completed – you would not put the roof up before the foundations are laid.

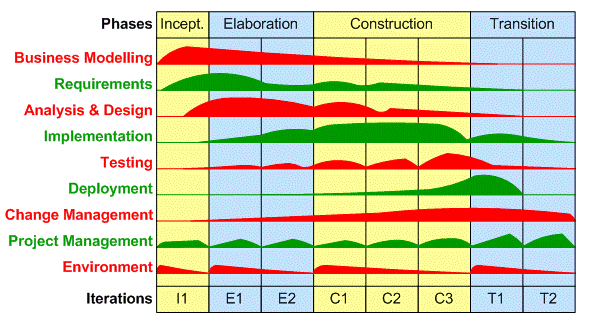
### Unified Process

Given the highly rigid nature of the waterfall methodology, this Graded Unit 2 project will adopt the Unified Process (UP). The UP complements the chosen development route (Java project using data structures) as agile methodologies allow for greater flexibility and transparency with the client, but still breaks development into four key phases: **Inception**, **Elaboration**, **Construction**, and **Transition**.

In each phase of the UP are tasks and artefacts already established that should be generated to evidence the goal or milestone that has been achieved. For example, the Inception phase generally involves the workflows Business Modelling, Requirements, and Analysis and Design. The Business Modelling workflow would include making a business case for the proposed system, and the artefact that should be generated is a feasibility analysis. The rest of the UP continues in this fashion, and the resulting primary deliverables like the feasibility analysis, requirements specifications, use‑case diagrams, project plans, amongst many other deliverables, result in accurate documentation for the project team to refer back to later in the project.

In general, an agile approach like the UP is far more suitable for software development projects as one of its key advantages is its iterative and incremental nature. There is no limit to the number of iterations (see Figure 5), so if a requirement is missed or it changes it is much simpler to revisit the earlier phases of the UP for revision. This makes the UP approach suitable for small or large development teams. Unlike the waterfall methodology, the UP is use‑case driven and so a greater emphasis is placed on the experience of the client and/or the end user. Overall, this results in a better-quality product that ensures business needs are satisfied. The UP sets out when and how Unified Modelling Language techniques for object‑oriented analysis and design will be used, which is an effective form of communicating the project’s requirements. This way of analysing and designing the system provides a structural support for developing the system’s structure and behaviour; the UP supplies the behavioural support.

Figure The UP is iterative and incremental by nature



*Figure taken from lecturer’s PowerPoint, Software Development Methodologies*

As flexible and versatile as the UP is, it has drawbacks that cannot be ignored. For example, it does not take into account any staffing, budgeting or contract management issues. There are six key workflow stages in the UP (Business Modelling, Requirements, Analysis and Design, Implementation, Testing, Deployment) and three supporting workflows (Change Management, Project Management, and Environment), so although the UP can work for a team of any size, it can be more complex for a small team to follow the overall model, particularly if there are team members who are not highly experienced which may result in staffing issues in the sense that the expertise is not available to achieve a key deliverable. For example, if no one in the team is a Business Analyst, it will be more difficult to fully understand how the system should complement business processes. Furthermore, there are often inter‑project issues in the sense that an employee may be working on a number of different projects simultaneously, but the omission of inter‑project work in the UP means that the opportunity for reuse in object‑oriented systems is missed.

Overall, the UP is the preferred methodology for this Graded Unit 2 project due to its agile nature. As senior software engineer Ricardo Balduino highlights, a streamlined version of the UP “preserves the essential characteristics of RUP [Rational Unified Process]”, which results in “a much simpler process that is still true to RUP principles” (Balduino, n.d.). The UP will bring me the key advantage of its ability to adapt to changing requirements, meaning if a requirement is missing later in the project it will be far easier to revisit earlier phases and ensure the end product is still successful.

#### Key Deliverables

As this project is adopting the agile approach of the Unified Process to develop the client’s new system, the typical deliverables produced throughout each phase to show a milestone has been achieved are outlined below.

| **Unified Process** | |
| --- | --- |
| **Phase** | **Deliverables** |
| **Inception** | * Business case made in the form of a **feasibility analysis**. * Vision Document in the form of an **Action Report** (this document) which sets the scope of the project. This includes: identification of primary functional and non‑functional requirements; initial top level use‑case diagrams; textual analysis. * **Project plan** that identifies: project scheduling; milestones; main tasks; resources required; and a Gantt chart. |
| **Elaboration** | * UML structure and behaviour **diagrams**. * An **executable** of a baseline version of the **evolving system**. This will act as the foundation for later iterations. * **Use case descriptions**. * **UI prototypes**. |
| **Construction** | * **Implementation** of the new system ready for beta and acceptance testing. * Potentially **documentation** that outlines **issue tracking** and **bug fixes**. |
| **Transition** | * An **executable** system * An **installation manual**. * **User manuals**. * **Support plan** for users. * **Plan** for **system upgrade** in future. |

### Summary

|  |  |  |
| --- | --- | --- |
| **Methodology Discussed** | **Benefits** | **Drawbacks** |
| **Waterfall** | * Requirements established very early * All project members know requirements and deadlines * Effective planning of time * Accurately estimates costs * Highly structured nature minimises changes to requirements | * One phase can’t begin before preceding phase complete * Extremely time consuming * Business needs and environment may change during development of project * Work in early phases may become irrelevant in time * Significant work if a requirement is missed * Little client interaction |
| **Unified Process** | * Agile – flexible and transparent * Accurate documentation * Easier to implement changing/missed requirements * Use-case driven – emphasis on end‑user experience * Satisfies business needs * UML effective in communicating project requirements | * Doesn’t consider staffing, budgeting, contract management issues * Nine workflows: complex for small teams to implement * Generally requires highly experienced staff * Inter‑project issues – reuse in object‑oriented systems important, but omits inter‑project work |

## Natural Language Analysis

The tables below reflects a natural language analysis performed on the initial project brief. I have avoided unnecessary repetition and have excluded any term that obviously refers to a location or name such as “Forth Valley”, and have excluded any term that obviously lacks meaning such as “like”.

### Potential Entities

The following entities (nouns) have been identified.

| **Entity** | **Summary** |
| --- | --- |
| 1. Independent law firm | Irrelevant to system but important for context – system being developed is for a law firm. Use of “independent” suggests it is a small firm. |
| 1. Case management system | The overall scenario. |
| 1. Employee | Refers to all client’s employees – Solicitors and Legal Secretaries. Potential superclass. |
| 1. Workload | Important in making a business case for the new system. System part of solution in controlling the firm’s workload. Potential package name. **Question client on maximum number of cases a Solicitor can handle at once.** |
| 1. Solicitor | A type of employee. Currently five Solicitors employed. Possible subclass of Employee. **Clarify with client what Solicitor details should be stored in system.** |
| 1. Legal Secretary | A type of employee. Currently two Legal Secretaries. Possible subclass of Employee. |
| 1. Case | A case is handled by a Solicitor. Potential for enumeration depending on implementation. |
| 1. Criminal law | Solicitors’ area of specialism – a Criminal Solicitor will handle a Criminal case. |
| 1. Immigration law | Solicitors’ area of specialism – an Immigration Solicitor will handle an Immigration case. |
| 1. Personal Injury law | Solicitors’ area of specialism – a Personal Injury Solicitor will handle a Personal Injury case. |
| 1. Specialism | The area of law specialised in by the Solicitor. Their specialism will dictate what type of case they can handle. Possible variable as some flexibility needed (specialisms vary). |
| 1. New case | Any employee should be able to create a new case. Likely to transform into a use‑case. |
| 1. Specific case | Any employee should be able to search for a specific case. Likely to transform into a use‑case. A valid id will be required. |
| 1. Details of a case | The system should store specific details of all cases. **Elaboration needed on specific case details.** |
| 1. Ability | Suggests potential use‑cases. |
| 1. Total open cases | Computed from number of cases in system’s storage. |
| 1. Unique reference number | Used to identify a case. **Elaboration needed regarding format.** |
| 1. Personal details | Data relating to a Solicitor’s client that should be stored in system. **Clarification needed from client regarding what details to store.** Suggest first and last name, DOB, address, phone number. Data will be entered by employees only. Possible instance fields. |
| 1. Client | Every Solicitor will have a client who a case pertains to. Possible class. |
| 1. Modestly specified equipment | The firm’s office equipment. |
| 1. Windows 11 | The Operating System used by the client. |
| 1. Company colours | The company colours – black, white, green; irrelevant to system functionality but potential for immutable class to set unchangeable RGB colour scheme. |

### Potential Actions

The following actions (verbs) have been identified.

| **Action** | **Summary** |
| --- | --- |
| 1. Allows their employees to control their workload | Irrelevant to system functionality but important for overall context; the objective of the system is to help in managing the firm’s workload. **Suggests system should be easy to use** as staff don’t have time to figure out the software. |
| 1. Specialise in | Refers to a Solicitor’s area of expertise. Suggests system should store this data relating to each Solicitor. **Required for action 5**. |
| 1. Employs | Refers to the firm employing two Legal Secretaries. |
| 1. Access to the system | Confirms all employees will need to access the system. **Elaboration needed on how they access system – will log in details need to be generated for staff?** |
| 1. Take on cases relevant to their specialism | Suggests system should **validate that a Solicitor is allocated a suitable case – whatever they specialise in as mentioned in action 2.** System should not allow a Criminal case to be assigned to an Immigration Solicitor. |
| 1. Create a new case | An ability all employees should have. Relevant details will have to be entered – client details, solicitor details, case details. **Elaboration needed on specifics of details to be entered.** |
| 1. View a specific case | An ability all employees should have. Will depend on input of case reference number. **Input will have to be validated. Elaboration needed on details to be returned when search performed.** |
| 1. Update the details of a case | An ability all employees should have. Will depend on input of a case reference number. **Input will have to be validated, as in action 7 above.** |
| 1. Delete a case | An ability all employees should have. Will require the system to provide functionality to remove all details relating to a specific case. |
| 1. Generate and view | All employees should have ability to generate and view reports. **Elaboration needed on potential report types.** Potential reports may be number of cases by type. |
| 1. Assign the unique reference number | When a new case is created, the system should assign a unique reference number to it. **Clarification needed on unique reference number format.** |

# Section 2 – Initial Planning Models

This section addresses the creation of the initial planning models. To reflect the chosen methodology of the Unified Process, this section includes an initial functional and non‑functional requirements list, and an initial top level Use Case diagram.

## Initial Functional Requirements

Having now interpreted the initial project brief, the basic functional requirements are identified below. Failing to satisfy these requirements will mean the system will not achieve its purpose.

1. Validate employee log‑in details.
2. Create a new case.
3. Search all cases.
4. Display caseload.

## Initial Non-Functional Requirements

The following non‑functional requirements of the new system are identified from the initial project brief.

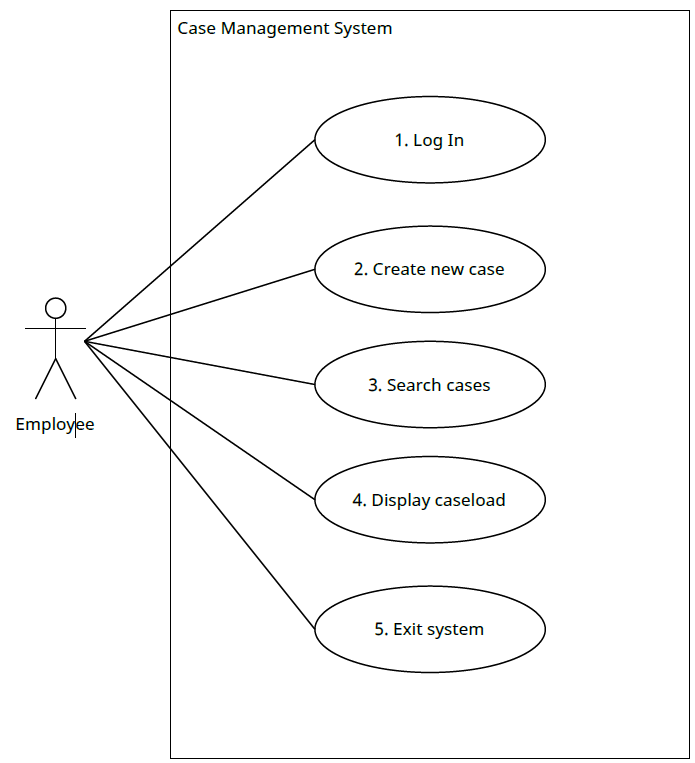
1. A case can be a Criminal case, an Immigration case, or a Personal Injury case.
2. A Solicitor can only take on cases relevant to their specialism.
3. Caseload information includes:
   1. total number of cases;
   2. total number of Criminal cases;
   3. total number of Immigration cases; and
   4. total number of Personal Injury cases.
4. The system should be styled using company colours – black, white, and green.
5. It should be designed to run on modest equipment running Windows 11.

## Initial Top Level Use Case Diagram

This use case diagrams aims to provide a bird’s eye view of the very basic functionality of the business processes involved in the evolving legal case management system. As described in Section 1, the Unified Process maps out the usage of UML throughout a project’s development. A use case diagram is a method of communication for those working in software and those with no technical background. A use case diagram is a method of succinctly conveying a “huge set of loosely defined requirements ... without losing important detail” (Miles and Hamilton, 2006). This is one great advantage of the Unified Process over the Waterfall methodology: if a use case is central to driving and developing the new system, then we are ensuring that we satisfy the requirements of the user. A use case, as described in O’Reilly’s *Learning UML 2.0*, is the fulfilment of one or more of your user’s requirements.

It should be stressed that this is simply the **top‑level** diagram, so this does not represent a definitive solution for the client’s new system. At this stage of the UP, I am simply establishing the most important functionality of the new system, being mindful that requirements can change and grow as the project advances. The functionality will be refined after information gathering techniques have been performed. However, this top‑level diagram will be presented and explained to the client who can then highlight in meeting 2 if we are missing any particular requirements in the system, and this consequently saves time and cost if this is highlighted during the Inception phase of the project.

Figure Top level use case diagram



It is evident in Figure 5 that the new case management system must, without exception, allow employees of the client to log in, create a new case, search for a case, view all cases, and lastly exit the system. These use cases will be refined and prioritised in Section 4 – Project Aims, and a variety of tests can be devised from the use case diagrams when the Transition phase is entered.

# Section 3 – Information Gathering

## Initial Client Interview

The initial planning models set out in the preceding section act as a good starting point to carry out an initial client interview. It is clear that there is some elaboration required in order to thoroughly plan this project. The client interview allows for clarification of what is required from the new system and information gathered from this interview along with other information gathering techniques is vital ***prior to*** preparing the final functional, non‑functional requirements, and fully dressed use case diagram.

The meeting agenda for meeting number 1 reflects the questions issued to the client prior to the first interview and is included in the following pages. Following this is the meeting minutes which summarise the key points discussed at the interview.

The justification for issuing the client with questions **as well as** a meeting agenda is to ensure there is written documentation of the discussions that have taken place. This is an effective form of capturing a written record of what functionality is especially important to the client in their new system. This will help when preparing a prioritised list of requirements as we are not relying on anyone’s memory when we are detailing these requirements – we can refer to this written record to ensure we capture details accurately. The client is likely to have a busy schedule, so preparing an agenda with specific questions helps in making the best use of the client’s valuable time. It also provides a clear direction for the interview, and the resulting documentation (the meeting minutes) will be very useful for future reference. This method of communicating is far more professional and effective than, for example, sending frequent emails back and forth to the client – this simply annoys clients and the validity of email responses cannot be guaranteed as the client may not read the email properly or may reply in a rush. It also results in disorganised correspondence if more than one project team member emails the client from different email addresses, which ultimately reflects badly on the team as a whole as it indicates a breakdown of communication. It is much more effective for all project stakeholders for communications to be monitored and tracked to ensure a consistent standard throughout the project.

### Client Questions

The following list of questions were issued to the client prior to the first interview.

1. Do staff currently have usernames to be used or would you like the new system to generate a new username?
2. Is there specific password criteria that must be met?
3. You mention staff should be able to create a new case. Can you provide us with the most important details of a case that must be entered?
4. Can you provide an example of case reference number?
5. Obviously every case will involve a client. Can you tell us specifically what client details you would like to store? For example, client’s first name, last name, DOB, address, phone number?
6. Similarly, every case will be handled by a Solicitor. Can you tell us specifically what details are to be stored for these employees? Should the system also store details about your Secretaries?
7. Should staff have the ability to add a billable activity? For example, if a solicitor has a meeting with a client, would you like to be able to record this information?
8. If you would like billable activities to be recorded, would you also like to be able to record the corresponding fee to be billed for that activity? For example, if a solicitor writes a letter, how much is charged?

Any information on your solicitors’ fees will help greatly.

1. Do your Solicitors have a maximum number of cases that they can be responsible for at any one time?
2. You mention employees should be able to search for the case reference number and view its details. What details would you like to be displayed when a search is performed?
3. Is there any other search functionality you would like us to implement?

We could implement, for example, searching for cases by legal area. This would mean you could view all Criminal cases at once, or all PI cases at once.

1. What is your budget and is this flexible at all?

Thank you for taking the time to answer these questions. We will contact you in due course with an update with regard to the progress of the project.

### Meeting Number 1 Agenda

The following agenda was also issued to the client prior to the first interview.

**Meeting 1 to be held at 10:00 on 25/01/2023**

**F1.130, Falkirk Campus, Forth Valley College**

|  |  |
| --- | --- |
| **Present:** |  |
|  |  |
|  |  |
| **Apologies:** |  |
|  |  |
| **Chair:** |  |
| **Minute Taker:** |  |

1. **Minutes of Previous Meeting**

N/A

1. **Matters Arising**

N/A

1. **The Log-In Process (<1 min)**

Log in details – current credentials or system to generate log in details?

Password criteria.

1. **Case Details (4 mins)**

Creating a new case – elaboration needed on most important details to be stored. Eg case reference number, case name, case type. Do all case types hold the same information?

Case reference number example.

Client details to be stored.

Solicitor details to be stored. Secretaries’ details too?

1. **Fees, Chargeable Time, and Caseload (3 mins)**

Recording billable activity.

Recording Solicitors’ fees.

Maximum caseload.

1. **Searching for Cases (2 mins)**

Other search functionality to be discussed.

1. **Ongoing Maintenance (<1 min)**

Budget.

1. **Date of Next Meeting (<1 min)**

To be confirmed.

### Meeting Number 1 Minutes

The following meeting minutes were prepared after the first client interview.

**Meeting 1 held at 10:00 on 25/01/2023**

**F1.130, Falkirk Campus, Forth Valley College**

|  |  |
| --- | --- |
| **Present:** | Daria Vekic, Project Manager (DV) |
|  | Susan Gardner, Client (SG) |
|  |  |
| **Apologies:** | None |
|  |  |
| **Chair:** | Daria Vekic, Chair (DV) |
| **Minute Taker:** | Daria Vekic, Minute Secretary (DV) |

1. **Minutes of Previous Meeting**

N/A

1. **Matters Arising**

N/A

1. **The Log-In Process**

DV asked details regarding the client’s log‑in process, specifically about current usernames. SG specified the new system should generate a username that would be suitable for this type of system, and welcomed for some example usernames to be proposed. **Action point: DV to prepare some example username formats and propose at next client meeting.**

A minimum password criteria was discussed and SG has specified that passwords should: contain a minimum of 12 characters, include at least 1 special character, include at least 1 number, and at least 1 uppercase character.

1. **Case Details**

DV pointed out that the project brief states staff should be able to create a new case and asked what the most important details of a case to be entered are. SG stated the project team can have some creative freedom here and welcomes evidence to support any choices. **Action point: DV to decide what details a case should store.**

DV asked the client to provide an example of a case reference number. SG answered with the following format: 001-CASE TYPE, so a specific example would be 001-CRI for a Criminal case.

DV also asked what client details should be stored, suggesting the system should store, at a minimum, the client’s first name, last name, DOB, address, phone number. SG added the system should also store enquiry type (such as Personal Injury enquiry).

Similar discussion around information to be stored about Solicitors. SG stated the same information should be stored for Solicitors but rather than enquiry type, should also store job title and hierarchy that they sit in, for example responsible for and to etc.

1. **Fees, Chargeable Time, and Caseload**

DV asked the client if they would like staff to have the ability to add a billable activity and gave the example of being able to record a Solicitor having a meeting with a client. SG stated they would like to have this ability but do not wish for monies to be taken in the initial build. SG also suggested costings would be helpful. **Action point: DV to set out various billable activities and costings.**

DV asked if Solicitors have a maximum number of cases that they can be responsible for at any one time. SG stated maximum caseload is 15 at any given time.

1. **Searching for Cases**

DV asked about details to be displayed when a search on case reference number is performed. SG replied all details about the customer and the case should be returned and displayed in a structured manner.

When asked about additional search functionality, SG noted the system should be able to generate a few reports but would like the project team to decide on a few specific reports to take forward.

1. **Ongoing Maintenance**

SG stated the budget for this project is £60,000.

1. **Date of Next Meeting**

DV will contact SG in due course to update on the progress of the project.

**Meeting concluded at: 10:15.**

### Summary of Client Interaction

The table below summarises the key points to be considered when refining the functionality of the new system.

|  |  |
| --- | --- |
| **Meeting No.** | **Summary** |
| 1 | **Log‑In Process clarified**   * System should generate a suitable username; examples will be proposed for client to choose. * Password criteria is: min. 12 characters, must include min. 1 special character, min. 1 number, and min. 1 uppercase character.   **Case Details clarified**   * Creative freedom regarding case details to be entered (see entity 14 of Natural Language Analysis and Summary) * Case reference number format. Format will be: 001-CASE TYPE. Eg. 001-CRI or 002-IMM(see entity 17 / action 11) * Client details to be stored: first name, last name, DOB, address (atomically split), phone number, initial customer enquiry so type of case can be categorised (see entity 18 / action 6) * Solicitor details to be stored: first name, last name, DOB, address (atomically split), phone number, job title, hierarchy they sit in (see entity 5 / action 6)   **Chargeable Time and Caseload clarified**   * Client wishes for additional functionality of adding billable activity, eg. a meeting with client. No monies to be taken in initial build. Costings helpful. (see entity 4) * Max. number of cases a Solicitor can handle at once: 15 (see entity 4)   **Search Functionality**   * Return all customer and case details and display in structured manner (see action 7) * Additional search functionality – generate few reports; project team to decide report type   **Budget**   * £60,000 |

## Background Research

It is important for all members of the project team to gain a fuller understanding of the importance and nature of case management in legal firms, and how it is critical to the efficiency and success of that legal firm, in order to develop a system that will satisfy the client. The client has specified that the project team can have creative freedom regarding what case details the new system should store (see minutes of meeting 1 with client). This background research will help in identifying exactly what data relates to a legal case and will also help in later phases of the overall project, particularly in the analysis and design workflows in the Elaboration stage of the UP, and in the implementation workflow in the Construction stage of the UP.

### General Job Responsibilities

The client has specified they currently have two Legal Secretaries. This role is sometimes referred to as Paralegal or Caseworker and typical responsibilities include, but are not limited to, the organisation of case files, writing drafts of legal documents, entering and analysing legal data, taking witness statements, and office administration (www.prospects.ac.uk, n.d.). This suggests that Legal Secretaries are very involved in the management of legal cases and an assumption that this user has poor IT skills should not be made. In fact, many Legal Secretaries may work part time alongside studying for a law degree, so it is likely that this end‑user is capable of learning to use a new system competently.

The end‑users of the new system will not be limited to the Legal Secretary role. The client has specified they employ five Solicitors: two specialise in Criminal law, two specialise in Immigration law, and one specialises in Personal Injury law. In general, typical responsibilities of all Solicitors include, but are not limited to, reviewing requests for legal advice from new and/or existing clients, offer legal advice in writing to clients, perusing case histories, reviewing evidence, meeting with clients, drafting legal documentation, and representing clients at court (www.prospects.ac.uk, n.d.). This suggests that the aforementioned responsibilities are potential billable activities which the new system may have to record for any type of case. **It would be helpful to clarify that this is correct in a second client interview**.

#### Immigration Solicitors

There appear to be minor differences in the responsibilities of a Solicitor depending on specialism. An Immigration Solicitor, for example, is likely to have to appeal Home Office decisions (www.law.ac.uk, n.d.), and that is unique to this specialist area. Immigration Solicitors often act for clients who are escaping dangerous political regimes (British Red Cross, n.d.) which raises the question of whether their client will have a permanent address or whether they reside in temporary accommodation. **It would therefore be helpful to clarify with the client if: (1) it is absolutely mandatory to input a client address when creating a new Immigration case (see minutes of meeting 1, agenda item 4), and (2) if Home Office decision appeal is a valid billable activity**.

In addition, Immigration Solicitors also provide assistance in making a variety of applications. More specifically, an Immigration Solicitor can help in making student visa applications, work visa applications, and visitor visa applications (London Immigration Lawyers, 2021). In making these applications, an Immigration Solicitor will often produce a detailed Letter of Representation which helps significantly in making a successful visa application. **It should therefore be queried with the client if this is a billable activity unique to Immigration Solicitors (see action point under agenda item 5 of minutes of meeting 1 with client)**.

#### Criminal Solicitors

Those specialising in Criminal law usually involves dealing with general crimes (www.law.ac.uk, n.d.), so it is possible that an important detail of a Criminal case which the new system should store is the crime committed (**see action point under agenda item 4 of minutes of meeting 1 with client**).

#### Personal Injury Solicitors

Further, Personal Injury cases generally involve acting for clients who have suffered an injury through an accident in the workplace or in public, or for clients who have suffered an injury through clinical negligence (www.law.ac.uk, n.d.). This suggests that Personal Injury cases may have subtypes such as: road traffic collisions, workplace accidents, work‑related illness, or accidents in public spaces. **It may be useful for the client if the user is able to easily select, perhaps from a dropdown list, what type of Personal Injury case is to be opened.**

#### Forms of Payment

It is apparent from the minutes of meeting 1 with the client (under agenda item 5) that there should be the ability to record billable activity in the new system. It is common for law firms to charge clients in one of two ways: either through an hourly rate, or through charging a fixed fee (AllAboutLaw, n.d.). **It should be clarified with the client if they charge by the hour or if they charge a fixed fee.** The ability to track billable activities is an important feature of the new system as billable activities are the firm’s main source of revenue. One aim of the new system is to help the firm to coordinate their workflow more efficiently so Solicitors are not wasting too much time on data entry and administrative tasks, especially when a lot of their working hours can be used on tasks like essential travel which generates no income but is still essential if, for example, travelling to meet a client in prison. In order for a legal firm to be profitable and for it to accurately project future revenue, it *must* be easy for the firm to record their billable activities easily and quickly. An article from The Law Society Gazette describes the “inevitable reality is that solicitors work long hours to account for the non‑billable work” (Wiel, October 2021). The accurate recording of income generated from one legal case, then, is extremely important to the overall success of the firm. This income will either come directly from the Solicitor’s client, or the case will be funded by Legal Aid. It is common for Criminal cases and Immigration cases to be funded through Legal Aid (www.lawsociety.org.uk, n.d.). Legal Aid is availability for certain Personal Injury cases, but not all (Irwin Mitchell, n.d.). **It would be beneficial to clarify with the client that their Solicitors’ fees are met either directly from their client, or via Legal Aid**.

## Investigation of Similar Systems

The initial client interview has brought to light some areas that require further research. We can see, for example, from the minutes of meeting 1 with the client that the project team have creative freedom to decide what details a case should store. Below is an investigation of similar systems currently used in many legal firms. The findings of this investigation will likely prove helpful in design decisions later in the project.

### iManage

iManage is currently used in many industries, specifically legal and accounting and financial services, for various management systems. It is reported that 2,500 law firms worldwide use iManage to “get work done by combining artificial intelligence, security and risk mitigation with market leading document and email management” (iManage.com, n.d.).

This particular software was used extensively in the writer’s previous employment from 2016-2021 in the context of an international law firm dealing with a variety of areas of law. The following evaluations from this anecdotal experience have been made.

#### Ease of Use

iManage was on the whole easy to use as it was integrated with a variety of Microsoft applications and implemented an user interface that was intuitive for users with competent IT skills, thus reducing the need for extensive training on how to use the overall system. To me, the user interface anticipated users' needs well as elements such as searching for a case number were easy to locate and understand, and recently accessed matters appeared in a tree structure on the left hand side of the screen. This made overall case management more efficient because searching for matters took only 2 or 3 clicks of the mouse.

#### Accessibility

The overall appearance of iManage's graphical user interface was not striking or aesthetically pleasing and it noticeably decreased the accessibility of their system. The visual design is one drawback of the system as the neutral colour scheme (see Figure 6) did not implement a high contrast and consequently some elements are harder to locate. This often led to moderate eye strain in many users when using the system for long periods of time. The colour scheme is a very simple factor in optimising software for accessibility and it does not take a huge effort to ensure a sufficient contrast ratio is implemented.

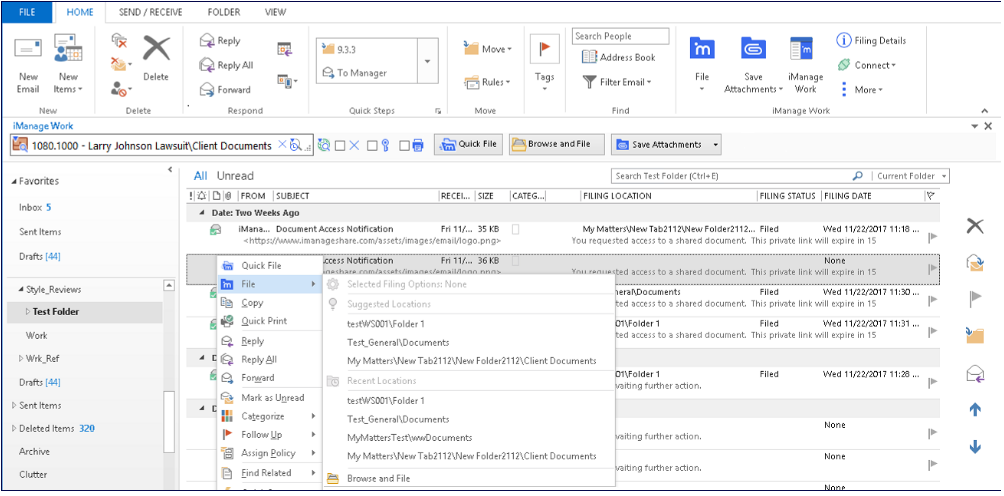


Figure iManage GUI

#### Functionality

If we sweep employees' eye strain under the carpet, though, it cannot be denied that iManage makes a remarkable difference in the way legal firms work. In terms of the functionality of their system, its performance is market leading. Their search functionality allows for searching of multiple databases that are integrated with their system. It also incorporates document management, allowing users to add, remove, view, open, move, send, lock, unlock, and sort document. In addition, it is extremely simple to create a new version of a document in iManage, and a convenient list of version history is easily accessible too, allowing the user to rollback to earlier documents in case any major error is made due to a lack of training.

Furthermore, the system offers convenient functionality to sort legal matters. Firstly, a list of recently‑searched‑for matters appear in a tree structure on the left side of the screen. To sort these matters, a user need only click "Activity Date" or "Name" from a dropdown list to sort recent matters by ascending or descending order.

Another interesting feature is the ability to add a matter to a user's "My Matters" list. This seems to be a similar feature to a "My Favourites" list in an ecommerce website, for example. It is easy to figure how this can make life easier for a Solicitor – if they have cases they are working on frequently, having the ability to add specific cases to something akin to a favourites list would save them some time and would require less clicks of the mouse to get to their desired case; almost like a speed dial but for legal matters. In this Graded Unit 2 project, the client wants to be able to manage their workload more efficiently. **It is perhaps a good idea to question if the client would like this feature implemented in their new system**.

### LawWare

Unlike iManage, LawWare is a practice management software used by UK solicitors. Its aim is to incorporate all aspects of managing a legal practice into a software product, and in doing so they are "setting new standards all designed to improve the efficiency" (LawWare Legal Practice Management Software, n.d.) of a legal firm. Its features are wide-ranging; some irrelevant to this Graded Unit 2 project such as legal accounting and bookkeeping. These features have therefore not been explored in any depth.

This software has also been used extensively in the writer’s previous employment from 2016-2021 in the context of a small, Glasgow-based law firm primarily dealing with Criminal, Private Client, and Family law. The following evaluations from this anecdotal experience have been made.

#### Fee Recording

LawWare allowed the user to record a solicitor's billable activities. This description of the activity would usually be dictated by the solicitor, but severe mumbling meant billable activities were often input inaccurately, and it was always the typist that got the blame for the solicitor's poor communication. I therefore think a more user-friendly software would allow the user to select a billable activity from a dropdown list, with the option of inputting extra text should the user wish to do so. This would not create any extra work on the user's part and would not waste time in having to revisit the case after clarifying with the solicitor what was actually said.

LawWare also allowed the user to input the corresponding fee. A similar issue was encountered here, as the relevant fee to be charged was either mumbled by the solicitor or ignored completely. It would be useful if the user could select the billable activity (from a dropdown list, as mentioned), and the corresponding fee box displays the correct fee according to the activity selected. This was a regular pain point for users of LawWare.

#### Search Functionality

The software's search functionality was generally fit for purpose. The relevant legal case would usually be presented when searched for, and the user would only have to double click on the desired matter to "enter" it. Users were then able to update the case as desired. There was no obvious way to sort any recently-searched-for matters as in iManage. There was also no ability to add legal matters to a "Favourites" list. On the whole, an accurate matter reference number had to be entered in order to retrieve the case.

### Summary

|  |  |  |
| --- | --- | --- |
| **System** | **Advantages** | **Pain Points** |
| iManage | * Integrated with Microsoft made it familiar and intuitive * User interface easy to navigate and understand * Searched for cases were displayed in a tree structured down the left side of the screen * Possible to integrate multiple databases * Effective document management * Document version history easy to view * Easy to sort matters by Activity Date or Name * Matters can be added to a "Favourites" list | * Poor colour contrast caused eye strain, made it less accessible |
| LawWare | * It allowed for recording of billable activities and corresponding fee * Search functionality retrieved correct case provided | * Accurate fee recording relied on verbal communication * No obvious sort functionality * No ability to add cases to a "Favourites" list |

## Meeting Number 2 Agenda

Having considered the background research and the similar systems investigated, a second client interview was arranged. The following agenda was issued to the client prior to the second interview.

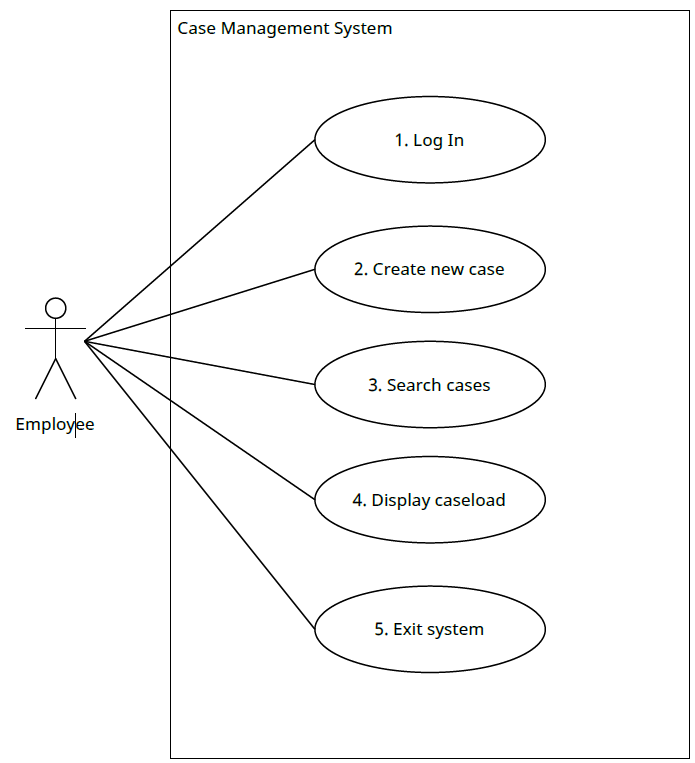
**Meeting 2 to be held at 09:30 on 08/02/2023**

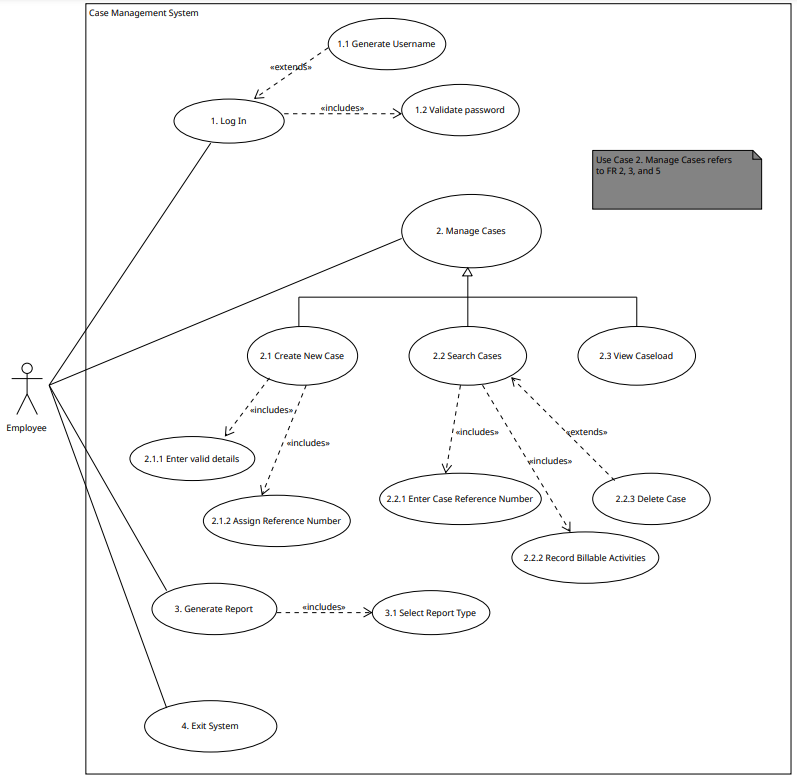
**Breakout Area, Falkirk Campus, Forth Valley College**

|  |  |
| --- | --- |
| **Present:** | Daria Vekic (DV), Project Manager |
|  | Susan Gardner (SG), Client |
|  |  |
| **Apologies:** |  |
|  |  |
| **Chair:** | Daria Vekic (DV), Chairperson |
| **Minute Taker:** | Daria Vekic (DV), Minute Secretary |

1. **Minutes of Previous Meeting**
2. **Matters Arising**
3. **Visual Aids**

The following diagrams represent what we think your employees should be able to do on the new system. Do you think our interpretation of your brief captures all of the important requirements of the new system?



**Refined Diagram:**

A conceptually ideal system will track and record information relating to all of the things identified in this diagram.

**CLIENT**

**CLIENT**

pertains to

pertains to

**CASE**

**CASE**

**CASE DETAILS**

**CASE DETAILS**

contains

contains

**CASELOAD**

**CASELOAD**

**USER**

**USER**

**REPORTS**

**REPORTS**

**EMPLOYEE**

**EMPLOYEE**

consists of

consists of

managed by

managed by

is a

is a

generates

generates

1. **Log‑In (<1 min)**

Example username formats:

1. staffFirstName.staffLastName (daria.vekic)
2. staffFirstInitial\_staffLastName (d\_Vekic)

Please kindly choose a format.

1. **General (3 mins)**

Home page – search bar immediately obvious to the user. Discuss potential dashboard design.

Navigation menu, visible at all times with the following buttons: Add New Case, View All Cases, View Criminal Cases, View Immigration Cases, View Injury Cases, Generate Reports.

Discuss "My Favourite Cases" list.

1. **Case Details (2 mins)**

When you search for a case, we propose to display the following details:

* Case Ref Num, Case Title, Case Type, Client Name, Solicitor Responsible

Clarify client details for Immigration clients.

Clarify Criminal case title format.

1. **Fees, Chargeable Time, and Caseload (2 mins)**

Discuss research on general job responsibilities of solicitors.

Propose to record the following billable activities for **all Solicitors**:

* First Letter to Client (offering legal advice)
* Perusing Case History
* Reviewing Evidence
* Meeting with Client
* Drafting Legal Documentation
* Hearing at Court

Propose additional billable activities for **Immigration Solicitors**:

* Home Office decision appeal
* Letter of Representation

Discuss how solicitors charge fees – fixed fee / by the hour.

Client payment – private or via Legal Aid.

1. **Date of Next Meeting (<1 min)**

To be confirmed.

## Meeting Number 2 Minutes

The following meeting minutes were prepared after the second client interview.

**Meeting 2 held at 09:30 on 08/02/2023**

**Breakout Area, Falkirk Campus, Forth Valley College**

|  |  |
| --- | --- |
| **Present:** | Daria Vekic (DV), Project Manager |
|  | Susan Gardner (SG), Client |
|  |  |
| **Apologies:** |  |
|  |  |
| **Chair:** | Daria Vekic (DV), Chairperson |
| **Minute Taker:** | Daria Vekic (DV), Minute Secretary |

1. **Minutes of Previous Meeting**

There were no issues highlighted in the minutes for meeting 1.

1. **Matters Arising**

No matters have arisen.

1. **Visual Aids**

DV presented the top‑level and refined use case diagrams (which can be found in Meeting Number 2 Agenda) to the client and explained that the use case diagrams represent the project team’s interpretation of what a user should be able to do in the new system. DV then asked if SG would like any other functionality to be incorporated into the new system. SG said the diagrams are a good and fair representation of the requirements of the new system and they capture the needs of the business accurately. **No amendments need to be made to these use case diagrams.**

DV then presented the conceptual model of the new system (which can also be found in Meeting Number 2 Agenda) to the client and explained that a conceptually ideal system will track and record information relating to the entities identified in this diagram. DV asked if this model captures the entities that are of significance to the business and SG is satisfied that the conceptual model captures the correct and relevant entities. **No amendments need to be made to this conceptual model.**

1. **Log‑In (<1 min)**

DV followed up on the action point under agenda item 3 and proposed the following username formats:

1. staffFirstName.staffLastName (daria.vekic)
2. staffFirstInitial\_staffLastName (d\_Vekic)

SG would like option 1 to be implemented in the new system. **Action point: DV to update requirements in Section 4 – Project Aims to reflect client's choice.**

1. **General (3 mins)**

There was general discussion regarding the search functionality of the new system. DV suggested that a search bar be immediately obvious to the user after the log in process and asked if SG would like this home page to be designed as a dashboard. DV explained the benefit of this would be that information such as current number of cases would be visible at a glance. SG advised that the **most important point is that the new system is as intuitive as possible as the overall objective is to make the running of the business more efficient but is happy for the project team to implement this.**

DV explained the intention of including a navigation menu that will be visible at all times in the system. It is intended that this menu will include the following buttons: Add New Case, View All Cases, View Criminal Cases, View Immigration Cases, View Injury Cases, Generate Reports. SG approves of this navigation menu. **Action point: DV to update requirements in Section 4 – Project Aims to reflect this detail.**

DV asked if the client would like a "My Favourite Cases" list and explained this would allow Solicitors to add a case to something that resembles a speed dial list. We could add this to the navigation menu. SG advised this is not a necessary requirement of the system and the aforementioned requirements should be prioritised, but if time and cost permits then this can be revisited in later iterations.

1. **Case Details (2 mins)**

DV drew attention to the action point in Meeting Number 1 Minutes under agenda item 4 and proposed the following details are considered to be the most important details of a case and when a search is performed, these details will be appear in a structured manner:

* Case Ref Num, Case Title, Case Type, Client Name, Solicitor Responsible

SG approved of this proposal.

DV summarised the background research which has been undertaken and asked for clarification regarding the client details to be accepted for Immigration clients. DV explained that it seems Immigration Solicitors frequently deal with asylum seekers and refugees, and therefore wanted clarification if an address is absolutely mandatory information. SG advised that all Immigration clients will have a registered address and will be input by a Solicitor or Secretary.[[1]](#footnote-1)

DV sought clarification of ambiguity relating to Criminal cases and explained it seems from background research they involve general crime. DV asked if a Criminal case title will differ to include the category of crime; an example would be **McRae v Dick – possession of illicit drugs**. The alternative is to standardise all case titles, regardless of case type, to read [client surname] v [opponent]; an example would be **McRae v Dick**. SG stated the firm's naming convention is the latter. **Action point: DV to update requirements in Section 4 – Project Aims to reflect this point.**

1. **Fees, Chargeable Time, and Caseload (2 mins)**

DV summarised the research undertaking concerning general job responsibilities of solicitors and proposed that the new system records the following billable activities for **all Solicitors**:

* First Letter to Client (offering legal advice)
* Perusing Case History
* Reviewing Evidence
* Meeting with Client
* Drafting Legal Documentation
* Hearing at Court

DV followed on to highlight that background research suggests an Immigration Solicitor may have billable activities unique to their job role. DV then proposed the additional billable activities for **Immigration Solicitors**:

* Home Office decision appeal
* Letter of Representation

SG confirmed that an additional billable activity for an Immigration Solicitor is Letter of Representation; Home Office decision appeal would be a brief description of a case. **Action point: DV to update Section 4 – Project Aims to reflect this detail.**

DV sought details on how clients are charged. SG stated their solicitors charge per hour and provided the following table with each solicitor's fees. SG also stated that they would like their employees to be able to specify how much time has been spent on a billable activity as this is how they keep track of monies owed to the firm.

|  |  |  |
| --- | --- | --- |
| **Solicitor Name** | **Specialism** | **Hourly Rate** |
| Mr Wullyum McRae | Criminal | £50 |
| Mr Java Duke | Criminal | £150 |
| Mr Gerry Butler | Immigration | £200 |
| Mr Jason Dom | Immigration | £200 |
| Mrs Suzanne Gardener | Personal Injury | £350 |

DV also explained that through background research it has been established that some clients pay for legal advice through Legal Aid. DV asked if the new system should allow the user to specify how funds will be paid. SG stated the system does not have to track this information and as long as it tracks the billable activities along with the time spent then their Secretaries will deal with invoicing the correct person/organisation.

1. **Date of Next Meeting (<1 min)**

To be confirmed.

# Section 4 – Project Aims

## Refined Functional Requirements

With some extra information now obtained through various information gathering techniques, it is now possible to refine the top-level functional requirements outlined in Section 2 – Initial Planning Models.

1. The system should validate employee log in details
   1. A username should be generated for new users
   2. A password must conform to specific criteria
2. The system should be able to create a new case
   1. Valid case details must be accepted
   2. Valid client details must be accepted
   3. Valid solicitor details must be accepted
   4. A unique case reference number must be assigned to the case
3. The system should accept a case reference number and display its details
   1. A case reference number must be in a valid format
   2. The system should remove a case when necessary
   3. The system should record billable activities entered by user
4. The system should display all cases when necessary
5. The system should generate suitable reports.

## Prioritised List of Functional Requirements

Having reviewed the refined functional requirements above, the list is already in a suitable order of priority. A justification is for each priority is provided below.

**Priority 1 – Very important**

1. The system should validate employee log in details
   1. A username should be generated for new users
   2. A password must conform to specific criteria

An employee logging in will be the first step in the overall user journey when the new system is implemented. If a user cannot log in, the system is meaningless and the client will not be able to access the software. There would be no point in undertaking this project if the system does not allow the employee to log in. If the system is not able to successfully generate a username for new users (and initially all employees will start as new users), then employees will not even be able to access their new system. Furthermore, the log in process should be made as secure as possible, so it should be able to successfully evaluate if the user’s password meets the specified criteria. Overall, if the log in process is not functional, then the system is not secure and the employees have no access to the system they are paying for.

**Priority 2 – Very important**

1. The system should be able to create a new case
   1. Valid case details must be accepted
   2. Valid client details must be accepted
   3. Valid solicitor details must be accepted
   4. A unique case reference number must be assigned to the case

If an employee cannot create a new case, then the system is unable to perform a basic CRUD function. The omission of this functionality would essentially mean the new system is simply a storage for data. The initial client brief refers to the new system as a case management system – the use of “management” suggests there should be basic CRUD functions available.

**Priority 3 – Very important**

1. The system should accept a case reference number and display its details
   1. A case reference number must be in a valid format
   2. The system should remove a case when necessary
   3. The system should record billable activities entered by user

Similar to the explanation above, a search functionality is part of the basic CRUD functions. The retrieval of relevant data is crucial to allow the user to then act on this data. For example, the user must be able to successfully search for a case before a case can be updated in any way or deleted from the system. The search functionality is the requirement that will be crucial in making this a user‑friendly and intuitive system to achieve the client’s overall objective of managing their workflow more efficiently. This requirement is not any less important than the preceding requirement (creating a new case) and so should be considered an equally important requirement of the new system.

**Priority 4 - Important**

1. The system should generate suitable reports.

This requirement is further down in the prioritised list of requirements because although it will help the client to view relevant data easily, it is not a functionality that is critical in the overall managing of caseloads. The ability to generate a report is just retrieving specified data and displaying it in a structured way that is easy for the user to read. Its main purpose is for employees to view information at a glance, but if the user cannot add or update a case, then one must question what worth is in generating reports if the information is not accurate.

**Priority 5 - Important**

1. The system should display all cases when necessary

This requirement is marked as Important because it will allow for easy navigation of the system. If, for example, an employee cannot remember an accurate case reference number, they will have the ability to view all cases at once and will offer them an alternative path to locating the relevant case.

## Final Functional Requirements

After prioritising the refined functional requirements, we arrive at the list below which can now be used as a baseline going forward. It must be acknowledged that agile methodologies do allow for changing and evolving requirements, so the term “Final Functional Requirements” should be taken loosely.

1. The system should validate employee log in details.
   1. A username should be generated for new users.
   2. A password must conform to specific criteria.
2. The system should be able to create a new case.
   1. Valid case details must be accepted.
   2. Valid client details must be accepted.
   3. Valid solicitor details must be accepted.
   4. A unique case reference number must be assigned to the case.
3. The system should accept a case reference number and display its details.
   1. A case reference number must be in a valid format.
   2. The system should remove a case when necessary.
   3. The system should record billable activities entered by user.
4. The system should generate suitable reports.
5. The system should display all cases when necessary.

## Final **Non‑Functional Requirements**

1. A username should be in the format: firstname.lastname (eg daria.vekic) (FR 1, FR 1.1).
2. A password must contain a minimum of 12 characters and include at least 1 special character, at least 1 number, and at least 1 uppercase character (FR 1, FR 1.2).
3. A suitable algorithm will evaluate if the password entered matches the user’s password (FR 1.2).
4. If log-in credentials are invalid (FR 1), the following message should be displayed:

Invalid log-in details entered!

1. A case can be a Criminal case, an Immigration case, or a Personal Injury case (FR 2).
2. Valid case details (FR 2.1) are:
   1. case type – either Criminal, Immigration, or Personal Injury;
   2. case title – [Client surname] v [Opponent]; and
   3. case description – a brief paragraph supplied by employee of reason for opening the case.
3. Valid client details (FR 2.2) are:
   1. first name, last name, date of birth, address first line, address second line, postcode, phone number, and initial customer enquiry type (such as PI for Personal Injury). Sensible algorithms should be used to validate details.
4. Valid solicitor details (FR 2.3) are:
   1. first name, last name, date of birth, address first line, address second line, postcode, phone number, job title, and job hierarchy (responsible for and to). Sensible algorithms should be used to validate details.
5. A case reference number must be in the format of:

*[three integers]-[first three letters of case type]*

For example: 001-CRI. It should be automatically incremented for every case that is added (FR 2.4, FR 3.1).

1. If a Solicitor is currently assigned to 15 cases, they cannot be assigned any more (FR 2).
2. A Solicitor can only take on cases relevant to their specialism (FR 2).
3. When a case reference number is searched for, it should display all details about the customer and the case in a structured manner (FR 3).
4. To remove a case, the user must click a Delete button (FR 3.2).
5. When the Delete button is clicked, the following message should be displayed:

Are you sure you want to delete this case?

1. A billable activity (FR 3.3) can be:
   1. First Letter to Client (offering legal advice)
   2. Perusing Case History
   3. Reviewing Evidence
   4. Meeting with Client
   5. Drafting Legal Documentation
   6. Hearing at Court
2. An Immigration Solicitor can have the following additional billable activities (FR 3.3):
   1. Letter of Representation
3. When all cases are to be displayed, it should display all details about the customer and the case in a structured manner (FR 4).
4. The reports that can be generated are (FR 5):
   1. overall caseload information including total number of cases, total number of Criminal cases, total number of Immigration cases, and total number of Personal Injury cases; and
   2. overall caseload information according to Solicitor.
5. The system should be styled using company colours – black, white, and green.
6. The system should be live by Monday 15 May 2023.
7. An interface which promotes ease of use should be implemented.
8. The system must implement security features.
9. The system should be designed using up to date technologies.
10. The budget is £60,000.

## Refined Use Case Diagram

Having considered all information gathered at this point in the project, it is possible to now produce a refined use case diagram. This is not indicative of the final project solution, but it does represent extra detail with regard to the system’s functionality and consequently allows us to determine the scope of the project as a whole which is helpful in ascertaining the feasibility of the project (Section 6 – Feasibility Analysis). A fully dressed use case diagram will be presented later in the Inception phase of the system’s development, but this (Figure 7) will be used as a baseline use case diagram moving forward as the client confirmed in meeting 2 (held on 08/02/2023) that it captures the most important functionality they would like in their new system.

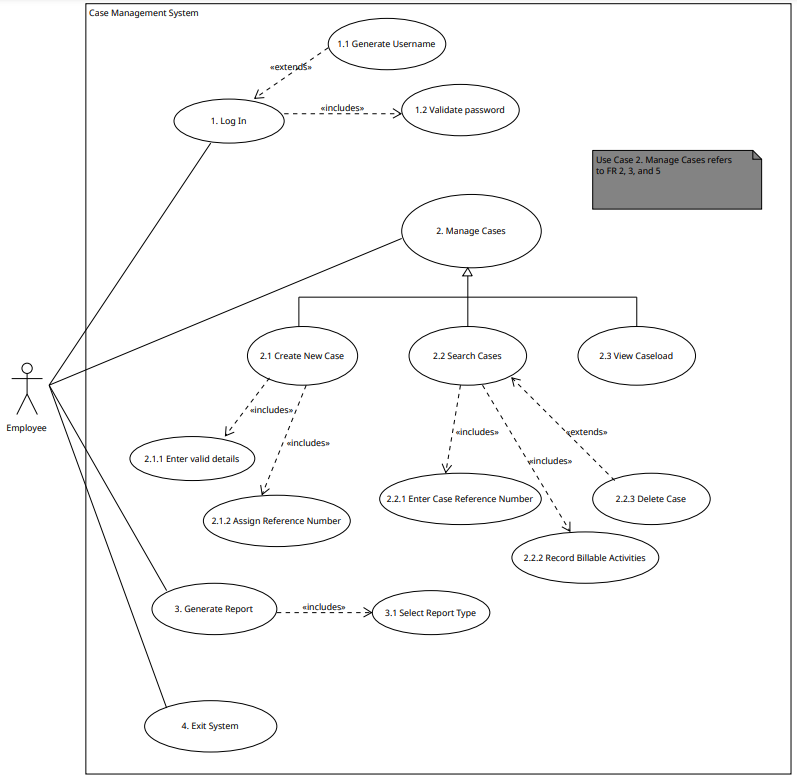


Figure Refined use case diagram

## Conceptual Model

It is important to create a conceptual model before charging ahead to build the client’s software. Similar to the purpose of producing a use case diagram, the conceptual model helps the project team to ensure that all the client’s users’ needs are captured. It is another way for a project team to leave no stone unturned. This in turn allows us to build a product that is fit for purpose and can be used with ease by the intended users. The client saves time and money as a result because the conceptual model addresses business needs, specifically what they envisage as being a conceptually ideal system.

The model on the following page is a top‑level conceptual model. This means that it identifies what we have interpreted – as a result of the brief analysis, initial planning models, and information gathering activities – as **important entities** to the client. It does not specify any attributes of these entities. In other words, its purpose is not to communicate the data which relates to these entities, but rather to ensure we are capturing data **about the correct things**. For example, the conceptual model on the following page does not show an entity called WAGES because in this particular context, it is not an important entity and the client has been clear that the new system is to *manage cases*. It would be possible to relate WAGES to EMPLOYEES (all employees receive a wage), but there is no suggestion from the Initial Project Brief or from client interaction (see Meeting Number 1 Minutes and Meeting Number 2 Minutes) that the new system should track employees' wages. Its primary purpose is to act as a visual aid to facilitate discussion with the client.

**CLIENT**

**CLIENT**

pertains to

pertains to

**CASE**

**CASE**

**CASE DETAILS**

**CASE DETAILS**

contains

contains

**CASELOAD**

**CASELOAD**

**USER**

**USER**

**REPORTS**

**REPORTS**

**EMPLOYEE**

**EMPLOYEE**

consists of

consists of

managed by

managed by

is a

is a

generates

generates

Figure Top-level conceptual model identifies important entities

## System Proposal

This system proposal is based on all information established so far in the Inception phase. It aims to confirm to the client that we have correctly interpreted the project brief and extracted the correct and most important requirements of the new system. It also aims to clarify supporting documentation included in the contract.

The objective of the new system is to allow for easier and quicker management of data relating to the client's caseload and clients which will allow for more efficient workflow coordination. The management of cases refers to the employee's ability to add a new case to the system, to update details of a case, to remove a case from the system, to record billable activities, and to generate reports. The new system is being developed for the overall management of case and client data and as such no sales or payment facilities are required in the initial build.

I propose to create a functional case management system that satisfies all of the above CRUD functions and meets the needs of the client's Legal Secretaries and Solicitors by Monday 15 May 2023 (N-FR 20). The system will be intuitive and easy to navigate with helpful information messages displayed when appropriate (N-FR 21). It will be developed as a Java program with data structures. The budget is £60,000 (N-FR 24) and includes the provision of supporting documentation, specifically installation instructions and a user manual. A breakdown of the system is provided below.

1. There will be a Log-In page for users to enter their credentials. The system will validate users' credentials (FR 1, FR 1.2). For new users, it will generate the user a username (FR 1.1).
2. If the log‑in credentials are valid, the user will be presented with the home page. This page will include a search facility which will be immediately obvious to the user. The search facility will allow the user to enter a case reference number to retrieve the desired case (FR 3).
3. There will be an easy‑to‑navigate menu visible at all times. This will allow the user to navigate the system quickly through the following buttons: Add New Case (FR 2), View All Cases, View Criminal Cases, View Immigration Cases, View Personal Injury Cases, Generate Reports (FR 4).
4. When the Add New Case button is clicked, the user will be presented with a form to enter required details (FR 2.1, FR 2.2, FR 2.3).
5. When the View All Cases button is clicked, all cases stored in the system will be displayed in a structured manner.
6. When the View Criminal Cases button is clicked, all cases belonging to Criminal Solicitors will be displayed in a structured manner.
7. When the View Immigration Cases button is clicked, all cases belonging to Immigration Solicitors will be displayed in a structured manner.
8. When the View Personal Injury Cases button is clicked, all cases belonging to Personal Injury Solicitors will be displayed in a structured manner.
9. When the Generate Reports button is clicked, the user will be able to select what type of report is to be generated.
10. When the user searches for a case reference number, the system will display the case details in a structured manner (FR 3). There will be a Delete button for the user to remove the case from the system (FR 3.2). The user will be able to click into the case and add billable activities or update case details (FR 3.3).

The new system will be coded as a Java program using data structures and will be tested thoroughly. I intend to implement a simple design using the client's colour branding (N‑FR 19) with a clear contrast between foreground and background colours. The aforementioned navigation menu will appear vertically on the left-hand side of the screen.

# Section 5 – Resources

This section provides an exhaustive list of resources that are required for the successful completion of this project. The resources identified here are the only resources available for use within the scope of this project.

## People

| **Resource** | **Justification** | **Cost (£)** |
| --- | --- | --- |
| 1. Project Manager | * Required for ensuring deliverables are produced on time and overall monitoring of project schedule * Responsible for managing project costs and resources * Key contact between project team and client * Nurture a positive team spirit | £40/hr |
| 1. System Analyst | * Required to analyse the business situation, identify opportunities for improvements, and to design the case management system | £45/hr |
| 1. Business Analyst | * Required to help the project team understand the business processes of the client. | £45/hr |
| 1. Developer | * Range of technical skills and competency required to construct the case management system | £55/hr |
| 1. Tester | * Required to test case management system satisfies functional requirements set out | £50/hr |
| 1. Stakeholders | * Includes all project team members along with the client * Client’s input very important throughout each key phase | N/A |
| 1. Client | * Required for project approval * Client input key to developing software that meets their needs | N/A |
| 1. Lecturers | * Required for emotional support | N/A |

## Hardware

| **Resource** | **Justification** | **Cost (£)** |
| --- | --- | --- |
| 1. Home computer | * Required for every stage of the project * Will be used for majority of project work * Cost N/A as already owned prior to project commencing * Device specifications: * Processor: AMD Ryzen 7 5700G with Radeon Graphics 3.80 GHz * Installed RAM: 8.00GB * Device ID: 1F47DAD1-21C5-477A-AED4-D4422FAF2836 | N/A |
| 1. Home Laptop | * Required for every stage of the project * Will be used for a lot of the project work, but secondary device to home computer * Cost N/A as already owned prior to project commencing * Device specifications: * Processor: 11th Gen Intel® Core™ i5-1135G7 @ 2.40GHz 2.42 GHz * Installed RAM: 8.00GB * Device ID: D7C84F5C-9B42-4377-8001-B8CA2FA9EB1D | N/A |
| 1. College computer | * Required to progress on project when in College * Accessed in timetabled classrooms * Modestly specified | N/A |

## Software

| **Resource** | **Justification** | **Cost (£)** |
| --- | --- | --- |
| 1. Windows OS | * Windows specifications for home computer: * Edition: Windows 11 Home. * Version: 22H2. * Installed on: 01/11/2022. * OS build: 22621.1105. | N/A |
| 1. Microsoft Teams | * Obtained and accessed through student profile. * Required for communication with client. * Required for communication with lecturers. | N/A |
| 1. Microsoft Word | * Obtained and accessed through student profile. * Required for production of documentation for this project. | N/A |
| 1. Microsoft Azure | * Obtained and accessed through student profile. * Student account obtained through lecturer. | N/A |
| 1. Microsoft Project | * Obtained and accessed through Microsoft Azure. * Required for the production of a formal Project Plan. | N/A |
| 1. Windows 11 | * The Operating System installed on personal machines. | N/A |
| 1. IntelliJ | * The intended Integrated Development Environment to develop the project in. | N/A |
| 1. Visual studio | * An alternative and unfamiliar IDE. * Accessed and obtained from https://code.visualstudio.com/ | N/A |
| 1. Eclipse | * An alternative, but not unfamiliar, IDE. | N/A |
| 1. Scene Builder | * Works with JavaFX to construct the graphical user interface. * Accessed and obtained at: https://gluonhq.com/products/scene-builder/ | N/A |
| 1. GitHub | * The construction phase of the UP requires good version control for any rollback activities. | N/A |
| 1. Google Chrome | * Browser required to access a variety of informational sources. | N/A |
| 1. Microsoft Edge | * An alternative browser required to access a variety of informational sources. | N/A |
| 1. Moqups | * Used for creating mockups which can be presented to client as a visual representation of what new system will look like. * Can be accessed here: https://moqups.com/ | N/A |

## Informational

| **Resource** | **Justification** | **Cost (£)** |
| --- | --- | --- |
| 1. Java 17 API | * The Application Programming Interface for Java. Will be used as a point of reference when programming the system. * Can be accessed here: https://docs.oracle.com/en/java/javase/17/docs/api/ | N/A |
| 1. Oracle Academy Java Foundations course notes | * These notes will be used as a reference point for topic areas relevant to the project. A reliable source for information relating to the basics of Java. * Accessed and obtained through Oracle Academy profile. | N/A |
| 1. Oracle Academy Java Programming course notes | * As above, but this source provides more in-depth detail relating to programming in Java. | N/A |
| 1. Oracle Academy Database Design course notes | * Required for general background information on potential development routes outlined in Section 1 – Assignment Brief Analysis. | N/A |
| 1. SQA Academy notes | * For general background knowledge on data structures. | N/A |
| 1. jetbrains.com | * It is necessary to explore an unfamiliar library or construct as part of this project. I intend to explore using a new IDE and will use this resource to download the IDE and also as a reference point for guidance on how to use the IDE. | N/A |
| 1. visualstudio. microsoft.com/ | * Another potential unfamiliar IDE. Should the project be developed in Visual Studio, this resource will act as a point of reference for guidance. | N/A |
| 1. https://docs.oracle.com/ javafx/scenebuilder/1/ use\_java\_ides/jsbpub-use\_java\_ides.htm | * It is necessary to explore an unfamiliar library or construct as part of this project. I intend to explore using JavaFX to construct the graphical user interface for this project. * This resource will help in designing the Graphical User Interface in JavaFX and also provides information on configuring Scene Builder in various IDEs. | N/A |
| 1. https://docs.oracle.com/ javase/8/javase-clienttechnologies.htm | * It is necessary to explore an unfamiliar library or construct as part of this project. I intend to explore using JavaFX to construct the graphical user interface for this project. | N/A |
| 1. Oracle Developers YouTube Page | * Contains informative videos on using JavaFX along with other relevant videos. * Can be accessed here: https://www.youtube.com/@oracledevs | N/A |
| 1. Oracle Learning YouTube Page | * Contains informative videos on graphical user interfaces, such as migrating Swing applications to JavaFX. * Can be accessed here: https://www.youtube.com/@OracleLearning/videos | N/A |
| 1. YouTube Tutorial, JavaFX – Create Banking Application with  Data Persistence – 2022 | * Potential personal project to explore using JavaFX. * Can be accessed here: https://www.youtube.com/watch?v=lkov5shwRQs | N/A |
| 1. Learning UML 2.0, Russ Miles and Kim Hamilton, O’Reilly | * For general background knowledge on Unified Modelling Language, used in the Unified Process. * Accessed through student profile. | N/A |
| 1. A Common-Sense Guide to Data Structures and Algorithms, Second Edition, Jay Wengrow ed. by Brian MacDonald | * For general background knowledge on data structures. | N/A |
| 1. GitHub | * The construction phase of the UP requires good version control for any rollback activities. * Contains general courses on how to use this version control software. * Can be accessed here: https://github.com/ | N/A |
| 1. MDN Web Docs | * The Transition phase of the UP involves delivering the executable system, an installation manual, and a user manual. I will use this resource to explore how to achieve these deliverables through developing a website. * Can be accessed here: https://developer.mozilla.org/en-US/ | N/A |
| 1. CodeAcademy | * Student subscriptions includes helpful courses that may prove helpful to this project. An example is their free course Learn JavaScript. Potential use in Transition phase. * Cost is N/A as subscription already owned * Can be accessed here: codeacademy.com | N/A |
| 1. W3Schools | * Contains fundamentals of a variety of Computing Science topics that may prove helpful in this project. * Can be accessed here: w3schools.com | N/A |

## Other

|  |  |  |
| --- | --- | --- |
| **Resource** | **Justification** | **Cost (£)** |
| 1. Kettle | * Required to boil water for the tea * Cost N/A as already owned prior to project commencing | N/A |
| 1. Cup | * Because you wouldn’t pour tea into a bowl * Cost N/A because already owned prior to project commencing | N/A |
| 1. Yorkshire Teabags | * Required to boost staff morale * Cost N/A because I'm not making the client buy me tea | N/A |
| 1. Milk | * Required to complete the tea * Cost N/A because I'm not making the client buy me milk | N/A |

# Section 6 – Feasibility Analysis

A feasibility analysis describes the feasibility – or viability – and risks associated with the project. This section of the report determines whether the project is even possible by investigation of the availability of factors such as time and resources. This report overall has so far documented the results of a variety of information gathering techniques (Section 3 – Information Gathering) and has also outlined the key deliverables expected at each phase of the project (Key Deliverables). The next sensible step in the Inception phase is to take all of this into account and evaluate if the project is realistically viable given the available resources set out in Section 5 – Resources.

## Technical Feasibility

It is established in Section 1 – Assignment Brief Analysis that the intended development route is to build the new system through a Java program that implements data structures. Having then identified the best methodology to adopt is the agile Unified Process, we know a number of deliverables will have to be produced in order to evidence a milestone has been achieved.

This obviously implies there will be a need to use appropriate technology. A suitable IDE will be required to implement the solution as a Java program with data structures. There are numerous IDEs that would be suitable for this project, such as Eclipse, IntelliJ Idea, Visual Studio, or NetBeans. The intended IDE to be used is IntelliJ Idea because it is known as the "Leading Java" IDE and is the preferred IDE by 3 out of 4 Java developers (JetBrains, 2019). There is a free edition of this IDE that can be downloaded from <https://www.jetbrains.com/idea/>. Obviously, a modestly specified computer with internet access will be required to access this IDE. Fortunately, the project team has access to two personal devices with suitable specification and also access to suitable equipment at Forth Valley College. Therefore, there should be no issue in accessing and obtaining this technology. Should this IDE prove to be unsuitable for whatever reason, the other aforementioned IDEs will be equally fit for purpose.

The Inception phase of the UP also requires a project plan to be produced. This means that a suitable project management software will be required. I do not foresee any issues in accessing Microsoft Project as this has been provided through a Student Microsoft Azure account at no cost to me through Forth Valley College.

It is also important to ensure the correct level of technical skills are available to complete the project. This is especially important to carry out iterations of the Elaboration and Construction phases of the UP as modelling business problems and building a successful system requires a good comprehension of how interfaces, abstract and concrete classes, subclasses, and enums interact with each other in Java. Thankfully, the project team have been studying Java for an entire 1 year and 3 months. In fact, our most experienced Developer has even completed the Java Foundations Learning Path through Oracle Academy. The Developer is also studying for an enhanced HND in Software Development, so the skill level required for the system to be developed is available in the project team (just).

Considering all of the above, the project to be developed is technically feasible.

## Financial Feasibility

Another important aspect of the feasibility of the project is the financial risk which the project may pose. The client has stated their budget is £60,000. The resources required to complete this project and their associated costs are outlined in Section 5 – Resources. The hardware and software that are necessary to complete this project do not have associated costs as the equipment and software have either been obtained prior to the project start date (18 January 2023), or is free to access.

The biggest cost of this project will be the people required to complete it. The project is scheduled for completion on Tuesday 9 May 2023, which allows for slippage time, so £50,000 of the budget is allocated to project team members for five months of work. We also need to make an effort to keep staff morale high, so we've allocated £100 for milk and Yorkshire teabags. This still leaves us comfortably within the client's budget, so if it becomes necessary to hire another team member, for example if someone in the current team leaves, then we will still have plenty of money to pay them.

Overall, the given budget is ample finance to complete this project.

## Time Feasibility

It is a well-known fact that to complete a project there must be the time to do it. The official deadline for the overall project is Friday 26 May 2023. However, to allow for slippage time, the schedule has been devised with an intended completion date of Wednesday 10 May 2023, as shown in the accompanying Project Plan. This allows for a realistic amount of time to rectify any problems that arise. For example, if testing activities highlight a bug in the system and the team need more time than scheduled to fix this, there is enough time to allow for this.

The schedule also gives a breakdown of the timings for each key phase of the development process. The Work Breakdown Structure (included in the Project Plan) breaks down the timings as such:

* Stage 1A – Inception Planning spans 19.29 days from 18/01/23 to 09/02/23;
* Stage 1B – Solution Planning spans 21 days from 09/02/23 to 08/03/23;
* Stage 2 – Development spans 45 days from 08/03/23 to 04/05/23; and
* Stage 3 – Evaluation spans 5 days from 04/05/23 to 10/05/23.

This scheduling takes into consideration that the most labour-intensive phases will be Planning and Development. These phases are therefore allocated extra time and this makes the overall timetable a realistic expectation of what can be achieved in the given time.

The schedule has been planned sensibly and as such there is enough time to complete a project of this scope.

## Legal Feasibility

If a software development project is legally feasible, then it will always adhere to any relevant laws and regulations.

Firstly, the system to be built is a case management system. There is no aspect of the new system itself that will break any laws or regulations. It will be built with security in mind, but ultimately it will be the responsibility of the client to ensure their employees handle sensitive data securely and confidentially.

However, any individual involved in the development of new software, particularly a software that is intended to organise data, must comply with various laws and regulations. If the project team come into contact with data belonging to any of the client's own clients, then the team will need to comply with the Data Protection Act 2018 which incorporates General Data Protection Regulation. This legislation is designed to protect everyone – it stipulates exactly how any individual's data is used and how it is protected, ensuring the rights of data subjects. A breach of this Act can result in severe consequences.

In addition, the project team must be informed of how to comply with the Computer Misuse Act 1990. This Act can be ambiguous in places as it can, at times, be hard to prove when access to a device or system is unauthorised. However, the project team should exercise caution at all times when working on a client's project. More specifically, screens should be locked when a computer is unattended and it is not encouraged to work on the client's project in public spaces. When constructing the new system, deliberately leaving a vulnerability in the code would be considered a breach of this Act.

The nature of the proposed project does not raise major legal feasibility issues. Rather, there are simply laws and regulations which the project team, and indeed all professionals working in Computing, must be informed of and must understand how to comply with.

## Summary

Having assessed the technical, financial, time, and legal feasibility of the proposed project, there is no evidence to suggest that this project cannot be undertaken. The most noticeable challenge of the project is the time available to complete it, but the schedule has been created to allow for slippage time so should extra time be required then the project is likely to still be completed comfortably within the deadline.

# Bibliography

AllAboutLaw (n.d.). *How does a law firm make money? | AllAboutLaw*. [online] www.allaboutlaw.co.uk. Available at: https://www.allaboutlaw.co.uk/school-leaver-law-careers/becoming-a-lawyer/how-does-a-law-firm-make-money#:~:text=As%20you%20would%20expect%2C%20law [Accessed 1 Feb. 2023].

Balduino, R. (n.d.). *Basic Unified Process: A Process for Small and Agile Projects*. [online] Available at: https://www.eclipse.org/proposals/beacon/Basic%20Unified%20Process.pdf.

British Red Cross (n.d.). *Asylum seekers: are they living on easy street?* [online] British Red Cross. Available at: https://www.redcross.org.uk/stories/migration-and-displacement/refugees-and-asylum-seekers/asylum-seekers-are-they-living-on-easy-street [Accessed 1 Feb. 2023].

Engebretson, P. (2013). *The Basics of Hacking and Penetration Testing : Ethical Hacking and Penetration Testing Made Easy*. Amsterdam ; Boston: Syngress, An Imprint Of Elsevier.

Irwin Mitchell. (n.d.). *How To Claim Compensation For Personal Injury*. [online] Available at: https://www.irwinmitchell.com/personal/personal-injury-compensation/guide.

JetBrains (2019). *IntelliJ IDEA*. [online] JetBrains. Available at: https://www.jetbrains.com/idea/.

LawWare Legal Practice Management Software. (n.d.). *Legal Practice Management Software - LawWare Legal software*. [online] Available at: https://lawware.co.uk/ [Accessed 5 Feb. 2023].

London Immigration Lawyers. (2021). *What does an immigration lawyer do? | Immigration assistance*. [online] Available at: https://london-immigrationlawyer.co.uk/what-does-an-immigration-lawyer-do/ [Accessed 1 Feb. 2023].

Miles, R. and Hamilton, K. (2006). *Learning UML 2.0*. Sebastopol, Ca: O’Reilly.

Wiel, L. October 2021. *The true cost of billable hours to the profession*. [online] Law Gazette. Available at: https://www.lawgazette.co.uk/commentary-and-opinion/the-true-cost-of-billable-hours-to-the-profession-/5110469.article.

iManage.com. (n.d.). *iManage delivers AI and knowledge management to all document management users with ‘Knowledge Unlocked’ solution*. [online] Available at: https://imanage.com/news-pr/imanage-delivers-ai-and-knowledge-management-to-all-document-management-users-with-knowledge-unlocked-solution/ [Accessed 4 Feb. 2023].

Wengrow, J. (2020). *A Common-Sense Guide to Data Structures and Algorithms, Second Edition*. Pragmatic Bookshelf.

www.prospects.ac.uk. (n.d.). *Paralegal job profile | Prospects.ac.uk*. [online] Available at: https://www.prospects.ac.uk/job-profiles/paralegal [Accessed 1 Feb. 2023].

www.prospects.ac.uk. (n.d.). *Solicitor, Scotland job profile | Prospects.ac.uk*. [online] Available at: https://www.prospects.ac.uk/job-profiles/solicitor-scotland [Accessed 1 Feb. 2023].

www.law.ac.uk. (n.d.). *Criminal Law | University of Law*. [online] Available at: https://www.law.ac.uk/employability/legal-practice-areas/criminal-law/ [Accessed 1 Feb. 2023].

www.law.ac.uk. (n.d.). *Immigration and Asylum Law | University of Law*. [online] Available at: https://www.law.ac.uk/employability/legal-practice-areas/immigration-and-asylum-law/ [Accessed 1 Feb. 2023].

www.lawsociety.org.uk. (n.d.). *Help with paying legal costs*. [online] Available at: https://www.lawsociety.org.uk/public/for-public-visitors/using-a-solicitor/help-with-paying-legal-costs.

1. Developer note: client address will therefore be mandatory for all case types and will require same validation. [↑](#footnote-ref-1)