Developing a database solution for a small wellbeing business to hold client data.

**Abstract:** The objective of this development project is to create an application for a business offering psychology and mindfulness service, allowing the business to store client information. The application will be using a document database, interaction with said database being handled through an API called through a front-end application. This solution aims to address the problem of managing the data for several clients within the business.

Project Description and Scope

The Problem

Information about clients is currently held by asking the client to fill out a physical pen and paper form Similarly, session notes are also written down as notes on separate sheets of paper. Such an approach has its own benefits during an initial start-up phase for a company. Most notably as such a method required little effort to integrate into the business’s workflow, hence providing a means for less technical business owners to easy hold client information for future reference.

This being said, storing personal data in physical forms brings about its own disadvantages, mainly in terms of security, redundancy, and ease of reference (lack thereof). Physical documents can be easily lost or damaged, either due to property damage or occasional clumsiness. Moreover, data may come in the wrong hands if there was ever to be a break-in in the business premises, or if the files holding such data were to accidentally be kept at an insecure location. Such concerns were made clear by the user, and from my personal experience when dealing with data using a similar method. Additionally, as the number of clients increases, searching for files relevant to the client during or in preparation for a session would in time become a tedious task requiring several human hours to achieve. Hence, the user in question requested a digitised solution to the problem.

Intended Project Output

Following interviews with the user, the application must be able to store all the information of a session for a client, which may also have a partner join the session for couple therapy. Multiple sessions for the same clients and aimed at addressing a particular behavioural change form part of an intervention. In case that a new behaviour needs to be addressed for the same client, this will create a new intervention, whereby a new group of sessions will address the new behaviour.

Additionally, sessions must be filtered through based on the client name and the behaviour that is being addressed. Upon selecting the session matching the criteria, the user must be able to view the information of that session, as well as invoking one of the following operations:

1. Create a continuing session for the same intervention as the session being viewed for the same clients(s).
2. Create a new intervention for the same clients(s) as the viewed session.
3. Create a new session for the same or different intervention for one of the clients in case of there being two clients.

In the coming months, the user is also considering running courses which would have a different structure to regular sessions. Setting up the application to handle the storage of these courses has been determined to be out of scope for the context of the project but may be added as an update at later stages.

As project output to this solution, I will be aiming to complete a fully functional MongoDB database which stores the required data, as well as developing an API using C# and ASP.NET Core to interact with the database. Additionally, a front-end application will be developed to allow the user to make use of these API calls from a more user-friendly application.

ICT Aspects to be Addressed

These aspects will be referenced throughout the paper as [Aspect x]

|  |  |
| --- | --- |
| 1 | Ensure that the most adequate database infrastructure is used for the storage and processing of client data and justify this choice. Module content focuses on using Jupyter notebooks and python to query and process data. My project will expand specifically on “*TM-351, Part 26: Applications of semantic web data*” in this regard, discussing how such data processing can be made accessible through an API. Adequate research must be carried out to apply this, adhering to the BCS (2019) professional issue 2a and 2b (see Appendix 2). |
| 2 | GDPR purpose (b) requires that data is only accessible by the people that need to make use of that data depending on their purpose (see Appendix 1). Due to this, the project will be expanding on *“TM-351, Part 12: Concurrency”* to discuss, not only how data can be accessed by multiple users at the same time, but also how role-based authorization can work in a NoSQL database. |
| 3 | GDPR principle (f) requires that backups for data is to be available in case of a system failure or a loss of data (see Appendix 1). TM-351 briefly discusses this but my project will further expand on the best way forward to back up a NoSQL database. |
| 4 | GDPR principle (f) requires that the database infrastructure is set up in a secure environment, and that tests are carried out to ensure that the data is indeed held securely and appropriately, ensuring that no loopholes can be taken advantage of to obtain. This builds up on “TM-351, Part 23 With data comes responsibility II: Keeping data safe” and adheres to the BCS (2019) professional issue 2c (see Appendix 2). |
| 5 | As the system must be future proofed and be able to handle several data from a number of clients, the operations that are run on the database must be as optimised as possible. Moreover, the API calls and responses must be developed in such a way as to not hinder performance. This aspect will be expanding on “TM-351, Part 16: Scaling out replication and sharding”. |
| 6 | Application must be able to create sessions as part of a continuing psychological intervention for a client (or clients) as well as creating new interventions for the same client. Data for sessions must be persistent, and this data must be filtered through allowing for the user to look up the client’s session information. This will require a front-end application. |

Project Planning and Lifecycle Model

List of Tasks and Associated Risks

**Tasks Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **(T)** | **Description** | **Risks** | **Time (Days)** | **(MS)** |
| 1 | Determine and organise required tasks. |  |  |  |
| 1.1 | Go through the provided module content to establish what is required from the project. |  |  | 1 |
| 1.2 | Structure required tasks in a table for easier reference as the project progresses. |  |  | 1 |
| 1.3 | Determine a realistic amount of time each task will take and justify. | R1 |  | 1 |
| 1.4 | Outline any risks that could be associated with each task and state mitigation for that risk. | R2 |  | 1 |
| 1.5 | Determine which project lifecycle would be best to implement into the project based on the tasks listed. | R3 |  | 1 |
| 1.6 | Create a schedule which reflects the chosen project lifecycle. |  |  | 1 |
| 1.7 | Update the tasks and schedule as more research is conducted and as the project progresses. | R4 |  | 1 2 3 4 |
| 2 | Conduct required research. |  |  |  |
| 2.1 | Revise module content that will relate to the project, namely related to: setting up a working database, choice of database system, transaction management, (de)anonymising data, legal and ethical issues. |  |  | 1 |
| 2.2 | Identify areas in the proposed solution which require further research and find and summarise papers which address these areas. | R5 |  | 1 2 |
| 2.3 | Cohesively make use of module content and knowledge from conducted research to justify the use of the technologies which will be implemented. | R5 |  | 1 2 |
| 2.4 | Identify any further areas that require research as the project progresses and carry out the required research. | R5 |  | 1 2 3 4 |
| 3 | Determine and obtain the resources and skills required |  |  |  |
| 3.1 | Review research notes to identify the skill and resources required. | R5 |  | 1 |
| 3.2 | Conduct further research to learn the skills needed for the completion of the solution if any. | R5 |  | 1 |
| 3.3 | Obtain any physical resources required (such as code libraries/hardware) based on the research conducted. | R6 |  | 1 |
| 3.4 | Identify any risks associated with the chosen skills and resources and state any mitigation required. | R2 |  | 1 |
| 3.5 | Review skills and resources as project progresses and update accordingly. | R4 |  | 1 2 3 4 |
| 4 | Determine most suitable technologies to implement solution. |  |  |  |
| 4.1 | Create ER diagrams to outline how the database will be structured and any operations which will be carried out on the database. |  |  | 1 |
| 4.2 | Create mock-ups for the front-end part of the solution to outline how the user will be interacting with the solution |  |  | 1 |
| 4.3 | Write down notes to establish how the front-end and back-end parts of the application will interact with each other. |  |  | 1 |
| 5 | Development of proposed solution (database architecture and api) |  |  |  |
| 5.1 | Create the required MongoDB database infrastructure that will be used by the application | R7 |  | 2 3 |
| 5.2 | Implement the database design that has been drawn out in task 4.1. | R7 |  | 2 |
| 5.3 | Insert dummy data which can be used during the solution’s development. |  |  | 2 |
| 5.4 | Develop an API using ASP.NET Core that will invoke commands on the MongoDB database. |  |  |  |
| 5.5 | Update database structure as the solution progresses based on feedback or any apparent changes that would improve its implementation | R4 |  | 2 3 4 |
| 5.6 | Push the database on a live environment |  |  | 4 |
| 6 | Development of proposed solution (front-end of application) |  |  |  |
| 6.1 | Create a working application with the technologies required. This will provide a framework onto which the views for the application will be built. Repository on GitHub should be set up for backup purposes. | R7 |  | 2 |
| 6.2 | Write the required functions that will interact with the database. | R7 |  | 2 |
| 6.3 | Create views/forms with the fields required that will allow users to interact with the database. | R7 |  | 2 3 4 |
| 6.4 | Update the front-end application based on user feedback or any changes to the database. | R4 |  | 2 3 4 |
| 6.5 | Proceed to push the application on a live environment |  |  | 4 |
| 7 | Address user feedback |  |  |  |
| 7.1 | Go through the application with potential users and list down any improvements or changes suggested. | R6  R4 |  | 2 3 4 |
| 7.2 | Go through tutor feedback based on the submitted assignments | R8 |  | 2 3 4 |
| 7.3 | List down these suggested improvements and state how these changes can be implemented and how they will improve the solution. |  |  | 2 3 4 |
| 7.4 | Implement the recommended changes. |  |  | 2 3 4 |
| 8 | Structure project report (TMAs/EMA) |  |  |  |
| 8.1 | Compile any relative resources that need to be submitted with the assignment into one document. |  |  | 1 2 3 4 |
| 8.2 | Review are assignment requirements and write up content required for the report |  |  | 1 2 3 4 |
| 8.3 | Review and reflect on project progress, stating what went wrong, what could go better, and how to progress moving forward. |  |  | 1 2 3 4 |

**Risks Table:**

[Back to tasks](#Tasks)

|  |  |  |
| --- | --- | --- |
| **(R)** | **Description** | **Mitigation** |
| 1 | Estimated time requirement for tasks may be inaccurate or certain difficulties or challenges may increase the time requirement of the task. | State mitigation for what may go wrong and how much time such a problem would take to solve. Construct a Gantt chart to visualize time for each task |
| 2 | Unforeseen risks | While creating the schedule. Leave some extra time for risks which come up that weren’t initially thought of. |
| 3 | As project progresses it may become apparent that the chosen lifecycle model may be taking too much time or isn’t ideal for the project. | Chosen lifecycle model can be structured in such a way that allows for it to be followed flexibly to account for this. |
| 4 | Too many changes may render the schedule obsolete drastically increase the time required for main task to be complete. | Consider the value of each recommended change and evaluate whether the time taken is worth it. |
| 5 | May not be able to successfully find research papers or module content that will provide an insight for the areas requiring research. Moreover, skills required for the chosen topics may require too much time to learn or read about. | Update the project scope based on the research conducted. Results in a smaller but overall more cohesive application. |
| 6 | Users’ availability may be limited due to their commitments and my own so their input may not always be readily available | Using the mock-ups and ER diagrams, discuss any future plans for the application so that the users can provide their input prior to the application itself being fully developed. Allows for fewer but more constructive discussions to take place. |
| 7 | Software may require certain prerequisites or installations that were not anticipated and come up during debugging. | Ensure that research is iterated through. Research must continuously take place in a development project using new technologies. |
| 8 | Provided feedback maybe misunderstood or focused on too much from my end, resulting in other areas being neglected. | Discuss any major feedback with the tutor and keep tutor updated on a regular basis. |

Project Lifecycle Model

Justification of using a **structured case model** at the start of the project

At the start of the project, the design will be mostly based off of my previous experience and existing knowledge of the problem and the technologies involved to implement the solution. Hence the solution is prone to be improved and expanded based on tutor and user feedback. Moreover, research may also dictate the approach of how the solution should be tackled as was discussed in risk 5. This results in an initial concept that will be frequently developed in terms of data collection, analysis, (re)interpretation and synthesis.

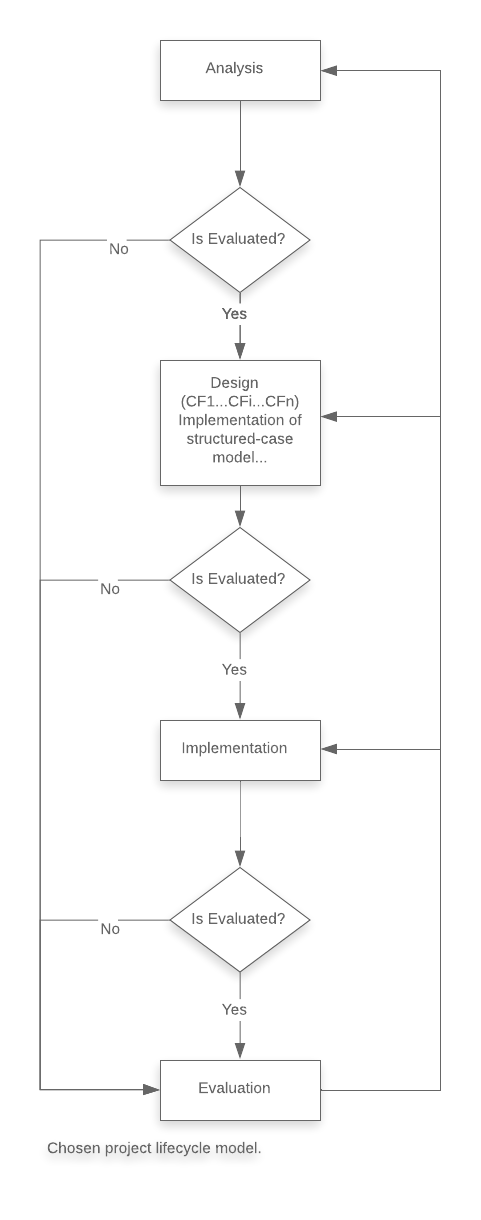
Using the structure cased model in the design step for the first two milestones ensures that afterwards, the conceptual framework for the project should be well established, which will then allow for development of the solution without frequent changes to the underlying knowledge for the solution taking place. Such an approach would allow for said knowledge to be developed without negatively impacting the overall schedule. Moreover, dedicating a sub-model for the design step of the overall model ensures that the chosen resources are scrutinised thoroughly, giving the solution a strong base for its implementation.

Justification of using a combination of **star model** and **no going back incremental model** for a combination of iteration and incrementation.

The solution requires evaluation to be based off of the application design as well as the analysis of the chosen problem rather than the developed implementation. Moreover, after the design of the proposed solution is expanded based on research, further evaluation must be conducted on the newly developed design, ensuring that the learning outcomes are met and the user requirements are satisfied, hence addressing the problem. This inherits from the star model where each step in the project lifecycle goes back to evaluating the progress of the project. This also holds due to the user feedback that will be provided throughout the project. After overall evaluation (which happens when each milestone is reached), the process must be reiterated to address this evaluation.

Since the project requires an application to be written from scratch and must be completed in a timely manner, both in terms of the database and the front-end aspect with which users will interact, a form of incrementation must also be present in the project. Iterating through processes frequently would be costly in terms of time and resources and would introduce the risk of not completing the tasks required for the milestones in time. Hence the project lifecycle will also incorporate aspects of the classic no going back project lifecycle model, whereby the project increments to the next step of the model after each evaluation of each step.

Both these requirements for my lifecycle model will result in a four-step plan starting from analysis, design, implementation and evaluation. Each of these steps will be evaluated once before proceeding to the next step, which will ultimately lead to the final round of evaluation before reaching a milestone in the project plan. Frequent evaluation from the tutor, the user, and myself will ensure that the solution will address the chosen problem appropriately. Afterwards, the model is then iterated once more using a more detailed conceptual framework to elaborate on these features, reflecting the literature review. This plan is outlined in the diagram.



Project Work Completed

Research to Determine Database to Use.

Considering Relational Databases

Relational databases enforce storing structured data, meaning that this data must adhere to a structure specified by the table before being inserted. For the proposed solution, two separate tables will be required to store client information and session information. Each client and session, stored as rows in the respective table, with each row storing the same data as other rows, making the use of a structured relational database a feasible option.

Moreover, relational databases allow for normalisation between the two tables. The client unique ID can be used as a foreign key for each session in the session table, allowing for multiple sessions to be linked to a single client row and in turn avoiding data replication of client data. [Aspect 2] To support multiple users, roles can be assigned to these users to make use of a relational database’s security schema, granting different levels of access rights depending on the user’s roles (The Open University, 2019, p. 4). Using the security schemas in this way complies with GDPR principle (f) (see Appendix 1)

[Aspect 3] Relational databases can be backed up by exporting the database as a backup (\*.BAK) file allowing for the backing up of Differential and Transaction logs, structures, data, constraints, and any stored procedures which may be used to invoke operations on the data. Moreover, several third-party software may be used to automate this process, satisfying the GDPR principle (f) (see Appendix 1) of backing up our data.

[Aspect 4] To further comply with principle (f) of the GDPR, as well as professional issue 2c (see Appendix 2) documentation online is extensively available to guide system engineers to implement a relational in a secure manner. Guyer C. (2019) goes through how to keep both the physical aspect of the database secure, as well as the necessary operating system configurations (mostly dealing with correct firewall setups), making use of roles, encryption, as well as naming numerous resources to assist in ensuring that the SQL Server is set up securely. I believe that, despite this being a blog, the credibility of the source is strong enough to consider the information provided. This is due to the fact that the blog is monitored by Microsoft, the creators of SQL Server, and so any misleading/inaccurate information will be removed, ensuring the validity of the provided information.

NoSQL Databases

Database can also be implemented by making use of aggregate tables. Aggregate entities contain all the data related to that entity (including sub-entities) in one aggregate. These aggregates will then be stored in a document known as a collection, whereby each database can consist of multiple collections. Moreover, similar to how JSON is structured, aggregate tables make use of key-value pairs to store the data for each entity. [Aspect 2] According to the MongoDB documentation (2019), this document database also supports collection-level access control by granting roles to users, hence only authorizing the users to access to invoke operations only on the required data. Making use of references and collection-level access fulfils the requirements set out by GDPR principle (f) (see Appendix 1) as well as professional issue 2c (see Appendix 2).

[Aspect 3] Backups in a document database can be set up by making use of replication. As also stated by the MongoDB documentation, replication copies the database onto different database servers, providing redundancy. Databases can be set up either as primary, secondary or arbiters. The primary database is the database used by the application and onto which operations are invoked. Replication copies the data into either a secondary database or an arbiter. In case of a system failure of the primary database, an election between the secondary databases and arbiters takes place to elect a new primary database from the secondary database, depending on how up to data the data is, ensuring that a primary database is always available. Implementing redundancy in such a away ensures compliance with GDPR principle (f) as well as professional issue 2c.

[Aspect 4] Guidelines set out by Guyer, C. (2019) in his blog regarding ensuring that the database infrastructure is set up securely can also be followed for when the document database is set up on as server in terms of the physical aspect. Such aspects include limiting access to the server room, setting up backups to be stored on a different site and limiting access to these backups. Moreover, some MongoDB specific features ensure that the database is set up securely. Murphy, D. (2017) discusses that MongoDB has five ‘core security areas’ namely being authentication and authorization, encryption, auditing and governance.

Authorization has been covered previously while discussing the purpose of roles in MongoDB. Authentication can be handled from the front-end application side as this isn’t natively supported by document databases or in this case, the community version of MongoDB. As discussed in the MongoDB documentation (2019), MongoDB can be configured to make use of SSL encryption, meaning that a certificate will be used to decrypt encrypted data passed between the application and the database, ensuring trust between the two parts of the application. Data can also be encrypted when stored and not just in transit (as is done using SSL) by using third-party software as suggested by Murphy. Auditing entails keeping a log of all the changes that took place in the database, allowing admins to identify any changes that took place to the database, allowing the removal of users who may be intruding, and to track down any harmful changes more easily to fix any issues that are caused due to these changes. Lastly, governance entails following a specified structure to maintain the integrity of the database. This can be done using Document Validation and its associated commands, allowing to define a schema that documents should follow. These security guidelines are in line with the security recommendations set out by the official MongoDB documentation (2019) and contributes to the fulfilment of GDPR principle (f) and professional issue 2c.

Justify Database to be Used

Hence, as discussed previously, an aggregate consisting of information regarding the type of intervention, client(s), and session notes would be a promising document database structure to consider. This is keeping in mind that the user of the application tends to have, on average, around 7 sessions per day, so the data store will gradually become rather sizeable.

The first justification for this decision is that, as was said in an *Introduction to NoSQL* (2012) retrieving a single aggregate (session with client information) has better performance as no joins are required. In a relational database, a more conventional approach would be to normalise the intervention, client information, and session notes into a separate table. Joining this data regularly (most notable in the case were data must be filtered) would hinder performance, as these join operations would have to take place for each session. As explained, normalisation reduces the risk of duplicating data. However, as Cunha et al. (2017) states, medical decisions are taken based on the client information that was available and accurate at the time, so data for those sessions should always persist. Hence, as was argued in the paper, any form of normalisation isn’t required in medical applications, so any links can be constructed as a single entity.

Additionally, *Introduction to NoSQL* (2012) states that a NoSQL DBMS has Improved flexibility over relational databases as no predetermined structure is set, further fields can be added in future by the client to store information as the system is tested. An example of this in our context is the addition of courses as a provided service, which will have a separate data structure from the previously explained sessions. The flexibility of NoSQL Allows for the developed API to remain unchanged, with the only difference being an addition of a filter to retrieve stored course or session information.

Word Count: 4007

References

Cunha, J., Fernandes, J. P., Ribeiro, H. and Saraiva, J. (2012) ‘Towards a Catalog of Spreadsheet Smells’, *Computational Science and Its Applications – ICCSA 2012*, vol. 7336, pp. 202–16 [Online]. Available from the OU Library at [http://link.springer.com.libezproxy.open.ac.uk/ chapter/ 10.1007/ 978-3-642-31128-4\_15](http://link.springer.com.libezproxy.open.ac.uk/chapter/10.1007/978-3-642-31128-4_15) (Accessed 20 February 2020).

British Computer Society [BCS, the Chartered Institute for IT]. (2019) *Code of Conduct for BCS Members* [Online]. Available at <https://cdn.bcs.org/bcs-org-media/2211/bcs-code-of-conduct.pdf> (Accessed 24 February 2020)

GOTO 2012 – Introduction to NoSQL – Martin Fownler (February 19, 2013) YouTube video, added by GOTO Conferences [Online] Available at <https://www.youtube.com/watch?v=qI_g07C_Q5I> (Accessed 1st March 2020)

Guyer, C., Milener, G., Hamilton, B., Kumar, S., Macauley, E., Rabeler, C., Hu, J., Matt., Laudenschlager, D. (2019) *Microsoft SQL Docs: Securing SQL Server* [Online] Available at <https://docs.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server?view=sql-server-ver15> (Accessed 26/02/2020)

Information Commissioner’s Office [ICO]. (2019) *Guide to the General Data Protection Regulation (GDPR)* [Online]. Available at <https://ico.org.uk/for-organisations/guide-to-data-protection/> (Accessed 24 February 2020).

Jan L. Harrington (2016) *Relational Database Design and Implementation, 4th Edition,* Morgan Kaufmann. Available at <https://learning-oreilly-com.libezproxy.open.ac.uk/library/view/relational-database-design/9780128499023/cover.xhtml> [Online] (Accessed 26/02/2020)

MongoDB (2019) *Introduction to MongoDB* [Online] Available at <https://docs.mongodb.com/manual/introduction/> (Accessed 27th February 2020)

Murphy, D. (2017) ‘The essential guide to MongoDB security’*, InfoWorld,* 3 February [Blog]. Available at <https://www.infoworld.com/article/3164504/the-essential-guide-to-mongodb-security.html>

Organisation for Economic Co-operation and Development (2002) *OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security*, OECD Council Recommendation, 1037th Session, 25 July.

Sánchez-de-Madariaga, R., Muñoz A., Lozano-Rubí, R., Serrano-Balazote, P., L. Castro, A., Moreno, O., and Pascual, M. (2017) ‘Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NoSQL approaches’, *BMC Medical Informatics and Decision Making*, vol.17, no.1 [Online]. Available at <https://bmcmedinformdecismak.biomedcentral.com/track/pdf/10.1186/s12911-017-0515-4> (Accessed 28th February 2020)

The Open University (2019) TM351 Part 14 Document Databases, Milton Keynes, The Open University

The Open University (2019) *TM351 Part 23 With data comes responsibility II: Keeping Data Safe,* Milton Keynes, The Open University.

The Open University (2020) ‘Legal, Social, Ethical and Professional Issues’, *TM470 Study Materials,* Milton Keynes, The Open University.

Appendices

Appendix 1: GDPR Relevant Information

GDPR Terminology (ICO, 2019):

**Personal data** will be stored in the application. This relates to data which can be linked directly to an individual, allowing users of this data to obtain information about these individuals and hence making said individuals identifiable. Individuals in the data set will be **identifiable** due to the data stored which allow the users to be distinguished from other users. Data held **relates to** the individual, not only because of the personal data held, but also because data relating to the activities of the individual is also stored.

System’s user will act as the **data controller.** Data controller determines what data should be collected, as well as what action should be taken by the user depending on the data which has been held by the client. In my context, there will be a single data controller which will be explaining the drawbacks of the current system and which problems they wish to address by using the proposed solution.

I will be acting as the **data processor** as. While developing the solution and in the context of my problem, what happens with the data is determined by the user, in hopes that the solution proposed will ease their workflow and futureproof their collected data. Moreover, I will not be collecting data myself in the final implemented solution, and all the data used will be dummy data to ensure that the application is working correctly. The outcome of the data won’t be of interest to me, as this outcome will be used by the user.

GDPR Principles (set out in Article 5 of the GDPR by the ICO (2019)). These must be regularly referenced to ensure the application is GDPR compliant, especially due to the sensitivity of the data. Chosen method must be able to allow the enforcement of the below principles. These are summarised versions of the relevant principles but will be referred to throughout the project to ensure that GDPR requirements are adhered to.

|  |  |
| --- | --- |
| Principle (a) – lawfulness, fairness and transparency | Use of the data collected must be justified and the processing of data must be lawful and won’t involve committing a criminal offense.  Data collected must help the individual it relates to and said individual must be made aware of the processing involved.  Be transparent in terms of what the data will be used for with the individual and the reasons for the use of this data. |
| Principle (b) – purpose limitation | Data processing must adhere to the initially established purpose for data collection. Any further purposes which are different from the initial purpose must align with the initial purpose and must have the individual’s consent. |
| Principle (c) – data minimisation | Data that is being stored must be sufficient to carry out the purpose set out initially and must also be relevant to that purpose. Data that isn’t relevant to the purpose shouldn’t be stored. |
| Principle (d) – accuracy | Where appropriate and adequate for the implemented solution, data should be kept accurate and up to date. Moreover, data should never be misleading, maintaining both the fairness to the individual the data relates to as well as the integrity of the data. |
| Principle (e) – storage limitation | Retention policies should be put in place to ensure that data isn’t kept any longer than what is needed to carry out the required purpose. The duration for which the data is kept should be justified, and data should be erased or anonymised when this duration is over. |
| Principle (f) – integrity and confidentiality | Data must be processed securely by means of “appropriate technical and organisational measures” which entails risk analysis, organisation policies as well as physical and technical assets. Data must be protected from unauthorized used and a testing process must be put in place to test security measures. Moreover, data should be available to individuals despite a system failure, hence backups are a requirement. |
| accountability principle | Ensures that processor takes the responsibility for the data stored and justifies all the above principles, ensuring that data is held appropriately. |

Appendix 2: Professional Issues

The BCS (2019) cited in ‘The Legal, Social, Ethical and Professional Issues’ (The Open University, 2020) states the following non-exhaustive professional issues that should be prioritised while working, planning, and conducting research on my project. These will be referenced throughout the project as justification for certain steps and measures in the project.

|  |  |
| --- | --- |
| **(PI)** | **Description of Professional Issue (PI)** |
| (2a and 2b) | Acknowledge the limitations of my knowledge and state what these are to determine if the planned project is a feasible one. Moreover, ensure that the resources available are also adequate for the work that is planned. |
| 2c | Ensure that the correct tools and techniques are used for the appropriate context. For my project, ensuring that the solution is built in a secure manner to avoid access from unauthorized users to the data. |
| 2d | Ensure that the software being used hasn’t been pirated and its use is justified and that I am entitled to be using. |
| 3b | Address conflicts of interest and how these can be resolved. I will be working while working on the project so state how I will be behaving professionally as both an employee and a student. |
| 3c | State how I will be taking responsibility in the case that the project won’t be successful by the end of the module. As I will be having a client, state how I will be handling this. |
| 3d | Ensure that data being used has the client’s permission for use. If not, state how this should be handled. |
| 4a | State if the project outcome will be fit for the purpose of the solution. Be honest, and if it doesn’t state why this is the case. |