



**UNIVERSITY OF WALES TRINITY SAINT DAVID**

# **Can an Agentic AI System Increase Willingness to Engage with Support Services in Wales?**

by

**Bradley Watkins**

A project submitted in partial fulfillment for the  
degree of M.Sc. Software Engineering (Web Development)

in the

**School of Applied Computing**

**Faculty of Architecture, Computing and Engineering**

Supervisor: Tim Bashford

January 2026



# Declaration of Authorship

I, [Bradley Watkins](#) declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed:

Bradley Cain Watkins

Date:

18/01/2026

# Form of Consent

I [Bradley Watkins](#) hereby consent that my Project, submitted in candidature for the M.Sc. Software Engineering (Web Development) degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed:

Bradley Cain Watkins

Date:

18/01/2026

UNIVERSITY OF WALES TRINITY SAINT DAVID

# *Abstract*

Faculty of Architecture, Computing and Engineering  
School of Applied Computing

M.Sc. Software Engineering (Web Development)

by [Bradley Watkins](#)

Accessing support in Wales is often hampered by a lengthy, confusing application process. Many are unsure of their eligibility or are nervous about how to start. This study aims to develop an agentic AI system capable of eligibility assessment, population of application form fields, and complete annotated PDF form generation. The system presents an easy-to-use conversational interface that employs a non-linear workflow for information gathering and follow-up questions.

The system uses Lang Graph's state graph framework, paired with custom tools and structured input and output via Pydantic models. A novel approach to agentic PDF manipulation is proposed, leveraging deterministic tooling and the large language model's core strength in information extraction. System testing demonstrated high performance with fields consistently populated successfully and documents annotated precisely.

Usability testing through a pre-post intervention study demonstrated a large effect size (Cohen's  $d = 1.02$ ), suggesting that the system meaningfully increased participants' willingness to engage. Unfortunately, while the system was effective in eligibility assessment and form generation, critical challenges were identified, such as hallucinations, user interface blindness, and systemic risks. The findings suggest that while the system shows promise in improving accessibility, there are fundamental risks to address before safe and trustworthy deployment is feasible.

# *Acknowledgements*

I would like to thank the DVLA for making this Masters possible, and Tim Bashford for keeping me on track.

# Contents

|                                                                  |             |
|------------------------------------------------------------------|-------------|
| <b>Declaration of Authorship</b>                                 | <b>i</b>    |
| <b>Form of Consent</b>                                           | <b>ii</b>   |
| <b>Abstract</b>                                                  | <b>iii</b>  |
| <b>Acknowledgements</b>                                          | <b>iv</b>   |
| <b>List of Figures</b>                                           | <b>viii</b> |
| <b>List of Tables</b>                                            | <b>ix</b>   |
| <b>1 Introduction</b>                                            | <b>1</b>    |
| <b>2 Literature Review</b>                                       | <b>4</b>    |
| 2.1 Agentic AI . . . . .                                         | 4           |
| 2.2 Agent Architectures and Design Paradigms . . . . .           | 6           |
| 2.3 Knowledge Representation and Graph-Based Reasoning . . . . . | 8           |
| 2.4 Retrieval Augmented Generation . . . . .                     | 10          |
| 2.4.1 Prompting . . . . .                                        | 12          |
| 2.5 AI Use in Medical and Form Filling Contexts . . . . .        | 13          |
| 2.5.1 Form Understanding and Manipulation . . . . .              | 14          |
| 2.6 Data Collection, Concerns, and Methods . . . . .             | 16          |
| 2.7 Ethical and Psychological Concerns . . . . .                 | 18          |
| <b>3 System Implementation</b>                                   | <b>21</b>   |
| 3.1 Architecture Overview . . . . .                              | 22          |
| 3.2 Multi-Agent Workflow Design . . . . .                        | 25          |
| 3.3 Knowledge Management and RAG Implementation . . . . .        | 29          |
| 3.4 PDF coordinate system challenges and solutions . . . . .     | 30          |
| 3.5 Session Management and Experimental Infrastructure . . . . . | 32          |
| 3.6 User Interface and Deployment Architecture . . . . .         | 33          |

---

|          |                                                                 |           |
|----------|-----------------------------------------------------------------|-----------|
| <b>4</b> | <b>Methodology</b>                                              | <b>35</b> |
| 4.1      | Experimental Design . . . . .                                   | 35        |
| 4.2      | Persona and Eligibility Mapping . . . . .                       | 36        |
| 4.3      | Sampling Strategy and Sample Size Justification . . . . .       | 37        |
| 4.4      | Data Analysis Plan: Measuring the Intervention Effect . . . . . | 38        |
| <b>5</b> | <b>Results</b>                                                  | <b>40</b> |
| 5.1      | Participant Demographics and Baseline Characteristics . . . . . | 40        |
| 5.2      | System Performance and Technical Metrics . . . . .              | 42        |
| 5.3      | Impact on Barriers to Support . . . . .                         | 43        |
| 5.4      | User Feedback . . . . .                                         | 46        |
| <b>6</b> | <b>Discussion</b>                                               | <b>50</b> |
| 6.1      | Lowering Barriers to Support . . . . .                          | 50        |
| 6.2      | The "Wait List" Paradox and Systemic Impact . . . . .           | 51        |
| 6.3      | Technical Friction as a Barrier to Trust . . . . .              | 52        |
| 6.4      | Autonomy vs Safety . . . . .                                    | 52        |
| 6.5      | Conclusion to Discussion . . . . .                              | 53        |
| <b>7</b> | <b>Limitations</b>                                              | <b>54</b> |
| <b>8</b> | <b>Future Work</b>                                              | <b>56</b> |
| 8.1      | Reliability and Verification . . . . .                          | 56        |
| 8.2      | Scalability . . . . .                                           | 57        |
| 8.3      | Knowledge Representation and GraphRAG . . . . .                 | 57        |
| 8.4      | Long Term Impact and Digital Exclusion . . . . .                | 58        |
| 8.5      | Systemic Issues . . . . .                                       | 58        |
| <b>9</b> | <b>Conclusion</b>                                               | <b>59</b> |
| <b>A</b> | <b>User Personas</b>                                            | <b>61</b> |
| A.1      | Elin . . . . .                                                  | 61        |
| A.2      | Sian Stewart . . . . .                                          | 62        |
| A.3      | Tom Davies/ Michael Davies . . . . .                            | 62        |
| <b>B</b> | <b>Support Forms and Eligibility Criteria</b>                   | <b>63</b> |
| B.1      | CHC Initial Screening . . . . .                                 | 63        |
| B.2      | Health Cost Exemption . . . . .                                 | 64        |
| B.3      | Local Mind Triage Request . . . . .                             | 64        |
| B.4      | Primary Health Support Referral . . . . .                       | 64        |
| B.5      | Specialised Mobility Aid Request . . . . .                      | 65        |
| <b>C</b> | <b>Agent Prompts</b>                                            | <b>66</b> |
| C.1      | Form Filling Agent . . . . .                                    | 66        |

---

|          |                                                            |                |
|----------|------------------------------------------------------------|----------------|
| C.2      | Eligibility Evaluation Agent . . . . .                     | 70             |
| C.3      | The Orchestrator (Main Chatbot Agent) . . . . .            | 73             |
| <b>D</b> | <b>JSON Coordinate Mapping for Form Fields</b>             | <b>82</b>      |
| D.1      | Json Schema for Deterministic Field Mapping . . . . .      | 82             |
| D.2      | Python Code to Map Filled Form to Annotated PDF . . . . .  | 98             |
| D.3      | Annotated PDF Generation Ready for User Download . . . . . | 102            |
| <b>E</b> | <b>Pre-Post Intervention Survey Responses</b>              | <b>106</b>     |
| E.1      | RAW CSV of Participant Survey Results . . . . .            | 106            |
|          | <br><b>Bibliography</b>                                    | <br><b>111</b> |

# List of Figures

|     |                                                                                                                                                                                                       |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | ChatState Definition . . . . .                                                                                                                                                                        | 23 |
| 3.2 | Flow Diagram Illustrating Multi-Agent Workflow Pipeline . . . . .                                                                                                                                     | 24 |
| 3.3 | Example JSON Output of the Agentic System’s Eligibility Assessment .                                                                                                                                  | 27 |
| 3.4 | Demonstration of rendering inconsistency: (a) defines the coordinate metadata, which results in divergent visual placements between (b) macOS Preview and (c) the Chrome browser environment. . . . . | 31 |
| 3.5 | User Interface Design . . . . .                                                                                                                                                                       | 34 |

# List of Tables

|     |                                                                         |    |
|-----|-------------------------------------------------------------------------|----|
| 5.1 | Comparison of Means for Paired Pre- and Post-Intervention Questions .   | 44 |
| 5.2 | Individual Participant Means and Effect Size for Paired Questions . . . | 46 |

# Chapter 1

## Introduction

Access to support services in Wales remains a significant barrier for many who could benefit from assistance. Despite the Welsh Government's dedication to delivering support, many experience difficulties navigating the application process. This is particularly true for those vulnerable people who may lack digital skills, confidence, or administrative knowledge required to identify and apply for relevant support.

The traditional approach to getting support relies heavily on static websites, printable forms, and human advisors that can be difficult to access. Several friction points emerge as people wade through support options that may not even be relevant, using difficult language and complex application forms [1]. These friction points have consistently affected those most in need of support, preventing vulnerable people from accessing help.

Recent advances in artificial intelligence offer an opportunity to address these barriers. Unlike traditional chatbots, AI agents can dynamically gather context and adapt to an individual's circumstances. Systems could assess eligibility across multiple support schemes, gather appropriate information from provided documents, and guide individuals through the application process. All of this can be done in a conversational manner, reducing the mental workload and improving accessibility.

The research presented addresses these gaps by developing and evaluating a multi-agent system for support service navigation. The system uses a graph-based architecture that orchestrates agents for conversational flow, eligibility assessment, and form field population. It aims to provide accurate, contextually relevant guidance while assuring that decisions remain transparent and explainable.

The primary research question is: Can an agentic AI system increase willingness to engage with support services in Wales? This is explored through several objectives:

- Design and implementation of a multi-agent system for eligibility assessment and form completion.
- Evaluation of the system's impact on user confidence and willingness to engage with support services.
- Identification of the technical and social challenges that must be addressed before such systems could be deployed.

The research uses a pre-post intervention study design with participants engaging with synthetic personas and support forms modelled on Welsh support services. This solution provides controlled evaluation of system impact while avoiding the ethical and practical complications of using real support forms and data. The study captures both quantitative measures of attitude change and qualitative insights into user experience and system limitations.

Three significant contributions are put forth.

- A novel coordinate-based PDF field mapping tool, addressing a prominent issue in programmatic PDF manipulation by AI agents.
- Evidence that a conversational agentic system can meaningfully reduce psychological barriers to seeking support in some groups.

- Identification of a critical safety and reliability concern that must be addressed before such a system can responsibly be placed in production.

The findings here suggest that while agentic systems show promise in reducing barriers to seeking support, engineers must carefully balance autonomy and user control. A foundation for future work in this domain is put forth, highlighting the importance of detailed evaluation alongside ethical considerations when developing agentic systems for the public domain.

# Chapter 2

## Literature Review

This literature review examines the current state of agentic AI systems, focusing on their use in healthcare and public services. The basic principles of agentic AI will be explored, followed by architectural considerations for these systems. The review will conclude with an examination of how agentic AI fits into public service, how it has been used to achieve related goals, and the ethical issues involved in producing and deploying these systems.

To understand the potential and limitations of agentic systems for support access, it is essential to define what an agent is and what sets it apart from traditional chatbots and linear workflows.

### 2.1 Agentic AI

Agentic AI stands as an important and influential evolution of standard conversational large language models (LLMs) [2]. Where LLMs were limited to a single context and sandboxed scope, AI agents are provided the autonomy and capability to complete tasks in a dynamic environment. This enhanced capability is typically achieved through a combination of natural language understanding and generation from LLMs, some level

of fine-tuning for specific tasks, and access to programmatic tools that the agent can use [3]. Models such as hierarchical language agents can manage multiple levels of decision-making, from strategic planning to detailed task execution, enabling effective handling of complex scenarios [4]. Pre-trained large language models typically serve as the underlying engine for planning, interpreting instructions, synthesising knowledge, and supporting iterative reasoning to refine responses [3]. Few-shot learning combined with prompt engineering, in which a model is trained or fine-tuned on a minimal number of labelled examples, can significantly improve outcomes [5]. In practice, many models can be trained for specific tasks with minimal upfront investment, provided the training data is high-quality.

The fundamental distinction between an LLM and an AI agent lies in the concept of agency, which is generally defined as an agent's capacity to achieve a goal autonomously. Unlike traditional rigid automation, agentic systems emphasise proactive and intelligent machine behaviour, helping them to manoeuvre complexity and unpredictability with or without human intervention [6]. However, agency is not a binary state, and clear definitions of the autonomy available to both the system and the human user are becoming increasingly important [4]. For systems operating in high-stakes environments, such as healthcare or support, the appropriate level of agency must be strictly defined to ensure user control and to maintain accountability [2].

In practical terms, an agentic system is designed to have a long-term goal and memory, allowing it to navigate interactions and correct its own path as necessary [7]. These features allow the agent to interact with its environment through actions and observations, forming an iterative loop that pushes beyond the shortcomings of classic conversational AI [8]. This iterative approach is important for complex, multi-step tasks, such as recommending and filling out forms based on a user's health profile [9].

While these conceptual foundations establish the theory, the practical implementation of these systems offers many options, each with benefits and trade-offs. The selected

design paradigms will have a significant impact on the system's capabilities and risks.

## 2.2 Agent Architectures and Design Paradigms

How agentic systems and workflows are designed is continually shifting as new technologies are introduced and research highlights novel ideas and anti-patterns. Current research is trending toward Multi-Agent Systems (MAS) over monolithic single-agent designs. MAS architectures offer a range of benefits, not least reliability and scalability, by allowing individual agents to handle parts of the overall context and to validate outputs. This will typically culminate in a hierarchical system in which a delegating agent determines what work should be assigned to its subject, or a coalition model in which certain agents are responsible for specific parts of the system. MAS, like all agentic workflows, faces considerable security challenges due to decentralisation and sociability, which can enable prompt injection by malicious users. Authentication, confidentiality, and secure design are absolutely paramount, especially when working with medical and highly sensitive personal data [8, 10, 11].

One of the main risks to privacy when working within an agent framework is the potential for hallucination due to the vast amount of data within an LLM's training pool [12]. This is especially concerning within the context of a support recommendation system, as there is a real risk of bias and erroneous responses in extremely sensitive scenarios. Despite efforts to identify and address these issues, it is critical that the system's design account for hallucination and guard against it.

The implementation of MAS has achieved great success in practice across many scenarios. One such implementation, ChemEagle [13], achieved an almost 50% improvement in text-based extraction and understanding over monolithic designs. They implemented a hierarchical system in which a planner agent oversaw specialised extraction agents and observer agents, whose roles were to provide feedback and identify failures. Although

this case is niche and not directly applicable, it highlights the benefit of having agents dedicated not only to the desired outcome but also to preventing an erroneous outcome [13].

Benefits have also been identified through a methodical approach, using structured prompts to collect the required parameters during interactions. One study [8] used a system of orchestrated, specialised agents to generate outlines, roles, patient briefings, and content for medical simulations. It is likely possible to use a simplified version of this to systematically collect user information, assuring a baseline understanding before recommendations are made within the system. The study in question used the n8n framework, leveraging proprietary LLMs such as Gemini 2.0 and GPT-4o, to successfully balance performance and cost-effectiveness. It highlights the value of remaining agile in the ever-changing landscape of AI and of preventing vendor lock-in through sensible technology choices [8].

At the centre of all of these technical and design decisions should be ethics and transparency. Current agentic systems are still prone to bias and hallucinated outcomes, so designs must mitigate risk and ensure the user remains in control. The precision and contextual consistency required to provide valid, medically relevant support recommendations to users are unlikely to be achieved solely through LLM integration. The literature has demonstrated a consistent limitation in LLMs in factual retrieval and logical inference, necessitating the integration of structured data frameworks [1, 9, 14].

These considerations highlight the need for structured, reliable knowledge representation to overcome the risk of LLM hallucination. Retrieval Augmented Generation and Knowledge Graphs have been key areas of research.

## 2.3 Knowledge Representation and Graph-Based Reasoning

Knowledge Graphs provide a mechanism for organising pieces of information into a structured, relational format, enabling more accurate, contextual, and precise reasoning than is possible with unstructured text. Recent research highlights that integrating LLMs with KGs is a significant trend, specifically to create systems that can leverage organised knowledge for complex tasks [15]. For instance, models like KRAGEN, which utilise graph-based methods, have demonstrated better performance than baseline GPT models and domain-specific LLMs like BioGPT [13]. KGs are inherently transparent in their reasoning, alleviating some of the black-box concerns of current AI and potentially allowing biases to be identified earlier, which is especially important in sensitive domains [9, 13, 16].

The construction of the necessary knowledge base includes various sophisticated techniques for extracting and structuring unstructured or semi-structured form data and text. Knowledge graphs (KGs) are semantic graphs composed of nodes and edges that represent knowledge about real-world objects. Graph Neural Networks (GNN) and Transformer-based models are used for domain-specific information extraction. Text is then transformed into a format, such as subject-relation-object triples, to allow mapping of dependencies [17, 18].

A knowledge graph is a way to represent real-world objects and their relations in a semantic way. Typically comprised of groups of triples, containing two nodes (real-world concepts or objects) and an edge depicting the relationship that links them. Refining knowledge graphs is vital to maximise their utility by resolving corrupt tuples, merging existing graphs, and normalising the semantics of relationships through the consistent use of ubiquitous language within the graph.

Construction of a knowledge graph requires several distinct steps. Entity discovery is the process of acquiring real-world concepts from input data, typing them into common concepts, and linking them to form a graph. The use of semi-structured data allows the system to receive hints for recognition and linking during construction, potentially disambiguating relationships that might otherwise be missed. Unstructured text provides no hints and relies solely on automated entity discovery and linkage. When providing unstructured text, it is essential to perform co-reference resolution to identify distinct texts that refer to the same entity, such as when an alias or pronoun is used in place of its given name. This is especially important during a refinement step, when distinct relationships can be identified as referring to the same concept, such as "date of birth" and "birthdate". If this linkage between nodes and edges is done without this refinement and background knowledge, the graph loses much of its semantic usability [19, 20].

The overall trajectory of KG research shows an increased focus on interpretability and qualitative evaluation [17]. This indicates an increasing need to validate KG models for practical, high-stakes, real-world applications, such as healthcare, where reliability is critical. This is especially important in a medical context, where unnoticed bias can be particularly insidious [21, 22]. The ability to map KG hops and reasoning into decision-making is a step towards explainable AI and provides users with a meaningful way to combat bias [23].

Typically, benchmarked knowledge graphs have been constructed from massive datasets. In real-world use, knowledge graphs will be built on smaller, domain-specific use cases. This could lead models to overfit due to a lack of nuanced, distinct data [24]. In this case, it is important to consider providing semantic hints or using semi-structured data rather than unstructured data during construction and training to improve performance [25].

Although GraphRAG has consistently demonstrated impressive performance across research and general benchmarks, it is important to note that it is not applicable in all scenarios. Existing naive RAG implementations have been shown to outperform GraphRAG on smaller datasets and with less powerful LLMs or SLMs (Small Language Models) [26, 27]. GraphRAG implementations have even been shown to fail entirely when integrated with SLMs under certain circumstances. There is also a cost concern, as large knowledge graphs require greater storage capacity and data duplication across both the KG and the standard database for RAG [26, 27].

## 2.4 Retrieval Augmented Generation

Retrieval-augmented generation (RAG) is another approach to making agentic systems more reliable and less prone to hallucinations and inference errors. RAG provides an external knowledge base to the LLM handling a given query, allowing it to enrich the query with further context before generating a response.

RAG enables the chunking, embedding, and storage of data in a vector store for retrieval by an external system. The LLM can then chunk and embed user input and perform similarity retrieval on the vector database. This allows the retrieval engine to determine the most relevant data for the query, which can then be injected into the LLM's context window. LLMs with access to RAG capabilities need less training on domain-specific information, thereby improving their accessibility and enabling faster iteration. With naive RAG implementations, it is observed that there can be difficulty with logical interpellation in connecting the original query to the given context; however, this can be addressed via antagonistic agentic systems, where another LLM is set up to judge the initial output [26].

This has become a standard practice for implementing agentic systems in complex contexts and has seen much success in many real-world deployments [8]. However, these

systems can suffer from inefficiencies in retrieval and difficulty with multi-hop reasoning, so care must be taken when implementing them in complex environments [23].

One way RAG-based agent systems have been enhanced in recent literature is by combining knowledge graphs and RAGs, also known as a KG-enhanced RAG system [13, 23]. An example of this is GraphRAG, which aims to achieve the benefits of both technologies in a highly complex industrial environment. They outline the requirements of chunking, relationship extraction, and knowledge graph generation from their existing data. During the retrieval phase, the agent system generates embeddings and keywords from the user input and utilises graph search algorithms via SPARQL and RDF Query Language [23].

The existing systems highlight a potential technical route to implementing the desired system. If the relevant data can be collated, generating a knowledge graph for available support will enable the utilisation of a KG-enhanced RAG system for form recommendations [23]. There may also be a possibility of utilising RAG for form filling once recommendations have been made, by extracting text from provided documents, processing it, and storing it in a temporary vector database. This database can then be queried to ensure that generated form data is relevant and contextually factual while keeping the agent's actual context window small [5]. Processing these documents could also help generate relevant queries for the recommendation system. Data protection and privacy will have to be carefully considered in this case, however, as preventing vectors for attack on users' sensitive personal data should remain the utmost priority.

For the capabilities of the knowledge representation architecture to be realised, it is essential that the system and the user can effectively communicate their intentions and desires. The quality of their interaction will depend heavily on the prompts and the strategy employed. By focusing on refining and constructing high-quality prompts, the system can achieve greater accuracy and contextual awareness, which is essential when using larger models without fine-tuning.

### 2.4.1 Prompting

An unusual software paradigm that improves LLM performance metrics is prompting [28]. The act of prompting in agent creation involves structured guidance or instruction for the model that serves as the basis for the agent's personality, understanding, and outputs. These prompts are textual inputs provided to the LLM as system messages, allowing the designer to give context and goals. The realm of prompting, or 'prompt engineering' as it has been colloquially termed, is one closer to the social sciences than an engineering discipline. Constructing a succinct and relevant prompt can be difficult, almost like guiding a junior or student. Still, prompting can yield effective performance gains and increase the relevance of the agent's output.

The development of effective prompts is a rapidly advancing area of study, and at this time it is difficult to pinpoint a single 'best' prompting technique [29]. This difficulty is only increased by the constantly evolving environment of LLMs, their parameters, and their ability to 'reason'. Studies have found that, for newer reasoning models, prompting can be detrimental to performance on simple tasks, acting as unnecessary noise. However, at this time, simpler non-reasoning models have been found to perform similarly or better than larger reasoning models on simpler tasks with fewer reasoning hops, allowing proven prompting techniques to be applied in these cases [30].

Generally, prompts have been distilled into four main aspects: profile, knowledge, planning, and reliability [28]. Crafting a prompt that conveys the agent's personality and goals, provides context, explains how it should achieve a goal, and offers assurances to reduce bias and increase reliability provides a powerful starting point. Trial and error is likely to be required from this point to focus the prompt and provide just enough domain and planning information to perform the given task well, without introducing noise or bias. One fundamental truth at this stage is the inherent unreliability of LLM output, and that any given prompt is likely to yield a diverse set of outputs. The ultimate goal,

then, is to structure the prompt so that these non-deterministic outputs reliably achieve the desired goal, providing confidence in their ability.

A promising prompting technique is that of Chain of Thought prompting [31]. Utilising a chain of thought, the output is such that the solution to a problem is accompanied by the steps that lead to that solution. This has been shown to improve arithmetic reasoning performance in larger models effectively and to demonstrate general applicability to more textual problem-solving as well. Beyond improving outcomes, chain-of-thought prompting may increase users' understanding of outputs and help move away from the black-box nature of many agentic chatbots and solutions. This has the potential to increase both trust and acceptance of agent interactions, an area that is uncertain at this time [32].

The following sections will examine agentic AI systems in the context of medical usage and form-filling, critical areas for this current research. Healthcare and support present unique challenges and opportunities for AI implementations, particularly around ethics and data privacy.

## **2.5 AI Use in Medical and Form Filling Contexts**

The medical field has historically been cautious about adopting deep neural networks due to their inherent lack of explainability [33]. Despite this, AI has great potential to transform paperwork in healthcare and other industries. Optical character recognition (OCR) technology allows paper documents to be scanned and digitised, creating searchable and editable versions. This can reduce manual data entry and simplify record management. For instance, systems like MEDLEE have been used to process radiographic reports and organise clinical data for easier access and analysis. Natural Language Processing (NLP) systems have also demonstrated effectiveness in extracting data from nursing notes and analysing breast pathology reports with high sensitivity and specificity [1].

However, challenges remain, with studies in the domain of form selection and completion facing barriers. Agents have been shown to struggle with understanding and editing PDF documents, especially those with deeply nested tabular structures. When converting PDF documents to other formats, much of the necessary formatting is lost, making understanding even more difficult for agents [1].

Another relevant study that attempted to complete paperwork with agents also faced substantial difficulties. The study used Visual Language Agents (VLAs) with access to photo-editing software via a graphical user interface (GUI). Current VLA and GUI agents have shown limited accuracy in form-filling tasks, primarily struggling with text placement and multi-step actions. Results saw a maximum completion rate of only 3.9% in their most difficult tests, when the agent was tasked with identifying field locations and overlaying text [14]. The results fall short of the necessary success rates but identify key findings that will support the implementation of this project. Text placement was a particular challenge for the agents, so it may help to provide structured information about the locations of form input fields through labelling.

An essential aspect of this research will be the generation of follow-up questions. One study used a competitive model comprising a question generator and a second model that attempted to answer the generated question using the available information. This allowed only appropriate follow-up questions to be passed to the user. The model showed promising results, and using such a system may alleviate frustration by avoiding the generation of questions the user feels have already been answered in their documents [34]. It highlights the importance of carefully designing antagonistic verifier agents within a MAS system to achieve greater reliability and precision.

### **2.5.1 Form Understanding and Manipulation**

Data extraction from forms is an ongoing and notoriously tricky area of research. The unstructured, typically non-computer-readable format of scanner-ready forms makes

them a prime candidate for both open- and proprietary-source solutions. Despite this apparent market gap for a solution, unstructured document extraction remains a veritable thorn in the sides of researchers and the community alike. This is especially true for PDF (Portable Document Format) files, which are non-machine-readable, contain sparse metadata, and are entirely unstructured and unpredictable in their formatting [35].

Many solutions have been developed over the years to address the problems posed by PDFs. Unfortunately, many of these fall short of the required success metrics to be viable in a production environment. There is also the issue of interoperability, as many of the solutions are tailored to handle documents within a specific domain [36]. Thus far, there are few successful methods for extracting semantic data from fully unstructured PDF documents.

Where extraction has been more successful, tools such as OCR (Optical Character Recognition) have been deployed, allowing the trained model to bypass working with the PDF directly. Instead, the PDF is reformatted or treated as an image, and data is extracted from it using machine learning. The extracted semantic information can then be mapped back to the PDF, or an ETL (Extract, Transform, Load) process can be used to present the data in another format.

Perhaps even more troubling is the manipulation, annotation, and editing of PDF documents. Despite mentions in magazine articles and the creation of code plugins for this task, there is little formal research in this area. The formal research that is there has typically taken one of the following approaches:

- Edit the source code of the PDF and recompile.
- Transform the PDF into an image and overlay text.

Both of these approaches, however, carry significant limitations. Editing the source code requires access to it, which is extremely unlikely in general and not possible for the use case in this study. Transformation and overlay could be a valid approach for this proof of

concept; however, it falls short if the PDF formatting and metadata need to be retained during form submission.

FormGym, in particular, provided its agentic framework with access to a browser-based photo editing software interface [14]. The agent then used the user interface to interact with and modify the PDF. The results were not spectacular; however, introducing friction as the agent navigated the human-tailored interface.

Open-source tools exist for manipulating free-text in PDF documents, such as PyPDF [37]. The use of such frameworks as tools for agent use is likely to be critical to the success of this project. PyPDF [37] in particular makes use of exact bounding box coordinates for placement of free text and other such annotations. Providing this information as semantic labelling is likely to dramatically improve performance over relying on an LLM or other machine learning models to determine these values. The labelling introduces a human step, which, from a usability standpoint, may create friction when moving the system to production.

The technical capabilities of agentic AI clearly show great promise; however, their implementation raises fundamental complications due to reliance on data quality and volume. Understanding the challenges of data collection in healthcare and the public domain is essential to designing effective systems.

## **2.6 Data Collection, Concerns, and Methods**

Ethical data collection, while not a new concern, is intensified by AI's capabilities. The 5Ps of ethical data collection are becoming increasingly difficult due to the black-box nature of many AI models. These 5Ps include [38]:

- Tracing data origins (provenance)
- Establishing effective security (protection)

- Preventing unintended data use (purpose)
- Creating accurate datasets (preparation)
- Safeguarding individuals from re-identification (privacy)

These difficulties arise because AI-driven data collection often falls outside current legal frameworks, despite ethical principles such as autonomy, justice, and trust being embedded in them [2].

Under EU law, data privacy is a fundamental right, and the GDPR is recognised as a leading data protection framework. The GDPR operates on an opt-in basis, requiring explicit, informed consent for the processing of personal data, which must be for a specific, legitimate, and necessary purpose. The law also mandates explainable AI (XAI) systems, with the expectation that users should understand the high-level logical processing and be able to request human intervention or review of automated decisions if dissatisfied [39]. Importantly, public posting of data does not imply consent for use, and profiling of sensitive attributes such as mental health or disability can be incredibly harmful. In many cases, anonymisation alone is insufficient due to modern re-identification techniques [40].

Implementation of AI in medical contexts requires not only initial training data but also a continuous supply for ongoing training, validation, and improvement [41]. Consequently, IT security safeguards will become increasingly vital to counter the risks of improper data use, inaccurate disclosures, and limitations in anonymisation techniques [4, 42]. Distributed machine learning (DML) offers a solution by enabling localised model training where only trained parameters, not raw data, are shared. Federated Learning (FL), considered highly promising, trains on local datasets on end devices, sending only the weights back to the server, allowing many users and data sources to contribute to training [38]. If transmission of sensitive data required for processing needs to be avoided, this is a worthwhile technology to understand and test.

The European Union Agency for Fundamental Rights has raised concerns about the quality of medical data and the risk of errors in AI applications. AI systems are often trained on idealised lab data rather than real-world medical practice data, which can introduce errors and bias when they shift to working with real data. Deploying AI systems for medical use based on faulty data generally violates the law, as most medical AI devices qualify as medical devices under the Medical Device Regulation (MDR). This is unlikely to apply to this project; however, considering the legal implications of deploying such a system for real-world use is vitally essential [43].

Data cleaning, while usually quality improving, can also introduce discrimination and bias into machine learning algorithms [43]. Therefore, high-quality data and effective data management at all levels are essential for machine learning in healthcare and related industries [44]. Informed consent should be obtained at every stage of the interaction, and users should have access to all data being processed and to the decisions being made. The produced model should provide explanations of where, in their data, specific information was derived to enhance explainability.

## **2.7 Ethical and Psychological Concerns**

User trust in AI is an incredibly complex topic, encompassing cognitive, affective, and behavioural components. Cognitive trust concerns the rational assessment of an AI's performance and predictability; consistent high performance fosters faith and a willingness to rely on the AI. However, users tend to trust AI less than humans inherently and may even be averse to algorithmic decisions, even when they outperform their human counterparts [45]. The affective aspect considers users' intellectual background and feelings towards AI, which can be less rational even if users are unfamiliar with AI. Behavioural trust develops over time from repeated direct interactions. Perceived usefulness and ease of use are key drivers in user trust, especially crucial in medical systems

[46]. Combining utilitarian and hedonistic, that being goal-oriented and joy-producing, behaviours can positively impact user trust [3].

Positive past experiences make users more likely to trust AI; however, negative experiences can just as easily destroy a user's trust in future interactions. A mismatch between user expectations and the reality of a system can also negatively impact trust. It has been found that those with higher educational backgrounds generally trust AI systems more than those with lower academic qualifications [46, 47]. This will be an essential consideration when designing a system for the Welsh context. One key concern among many appears to be confidentiality, with some also raising concerns about the costs AI systems may incur [47]. Some studies have shown that providing users control over their data, explaining AI systems, and explaining how data is used can increase confidence [48].

Especially for an AI system focused on supporting individuals, emotionality and human likeness will be important considerations. Users of the system may be in sensitive situations, and the system failing to meet their emotional needs may lead to a decrease in trust. Balancing the psychological and emotional language provided by the agent will be essential to avoid a mismatch with the user's expectations [3]. Studies have shown that younger people are generally more accepting of AI technologies, especially in a medical context [49].

When designing emotional interactions into the system, however, risks must be considered. AI systems, by their very nature, are not empathetic and cannot truly understand the emotional situation that they are responding to. Strict guidelines must be put in place to prevent escalation and misunderstandings if conversational responses are allowed. This is especially true when the system has access to a large corpus of the user's personal and medical data, which could be used in a response. Some researchers argue that emotional AI technologies should be strictly regulated due to their ability to influence decision-making and their potential for deception [50].

Another essential aspect to consider will be the design of the user interface and the application's front-end. Accessible interfaces are critical to digital interactions, and system efficiency is directly related to interface efficiency. Anthropomorphic elements, such as avatars and pop-ups, have shown potential to improve user acceptance and perceptions [51, 52]. Ultimately, the success of these systems depends on their ability to benefit the user at minimal cost, be that financial or mental [53].

# Chapter 3

## System Implementation

This section outlines the core requirements for the agentic system and the rationale behind its architectural design decisions. The system must enable the following core functionality:

- Autonomous eligibility assessment
- Accurate form field population
- User-guided document generation
- Seamless conversational integration across workflow stages
- Transparency and reliability

These requirements will help to address key barriers identified in the literature. These include navigating complex support services, ambiguity around eligibility, and technical challenges with document manipulation [1, 14]. To satisfy these requirements, the system employs a multi-agent architecture integrated with a graph-based workflow engine. This is supported by robust prompts for agent behaviour and state management protocols that ensure complete context is maintained.

The design ensures adaptability and agency in responding to user needs while maintaining user control in the process. The aim is to reduce the administrative burden on users when navigating the initial stages of support-seeking. The following sections detail the implementation approach taken to achieve these requirements.

### 3.1 Architecture Overview

The system is built on a graph-based architecture that leverages LangGraph's [54] StateGraph framework to manage multi-step workflows in agentic systems. Unlike traditional linear pipelines, the StateGraph architecture enables routing between specialised processing nodes. Each of these nodes is intended to manage individual functionality within the pipeline, allowing greater separation of concerns and control over workflow steps. When integrated with agentic AI, the routing can be done based on user intent and conversational flow. This enables powerful, non-linear workflows that more closely mimic real-world interactions.

At the core of the architecture lies the ChatState management pattern, implemented using Python's TypedDict and LangGraph's message-accumulation functionality. This state management technique retains a comprehensive conversation context throughout the entire user session, including message history, eligibility results, form data, and processing metadata. The ChatState utilises the `add_messages` annotation to automatically handle message accumulation, ensuring that no conversational context is lost as users make additional requests. By using ChatState to manage context, much of the maintenance overhead and risk of losing state in an agent's context window is alleviated. ChatState also aligns with the system's dynamic nature, allowing users to interact and provide information in any order while the context is iteratively updated. The state can also be passed to any node in the flow, ensuring that all decisions have access to the full context to provide meaningful output.

```
1      class ChatState(TypedDict):
2          messages: Annotated[list[BaseMessage], add_messages
3
4          ]
5
6          next_step: str | None
7          eligible_forms: list[str] | None
8          eligibility_results: dict | None
9          evaluation_complete: bool
10         filled_form: FormSchema | None
11         document_context: str | None
12         info_request_count: int
13         completed_forms: list[str]
14         current_form_name: str | None
15         missing_fields: list[str]
16         output_paths: list[str]
17         session_id: str
18         uncertain_forms: list[dict] | None
```

FIGURE 3.1: ChatState Definition

The system's workflow is made up of six specialised nodes, each with a core functionality:

- chatbot: for handling user interaction and decision making.
- tool\_node: that determines which node to call when the chatbot invokes a tool.
- evaluate\_eligibility: which passes conversation context to a specialised agent for evaluating support eligibility.
- process\_eligibility: that parses the eligibility agent's output in preparation for use by the chatbot node.

- `populate_form`: which passes conversation context to a specialised agent for the extraction of user information to populate form fields. If information is missing, follow-up questions will attempt to fill in the blanks.
- `generate_pdf`: that parses the output from `populate_form` and manages actual document annotation and generation using PyPDF.

The following flow diagram illustrates the workflow pipeline, how information moves through the system, and the system's ability to navigate in a non-linear manner.

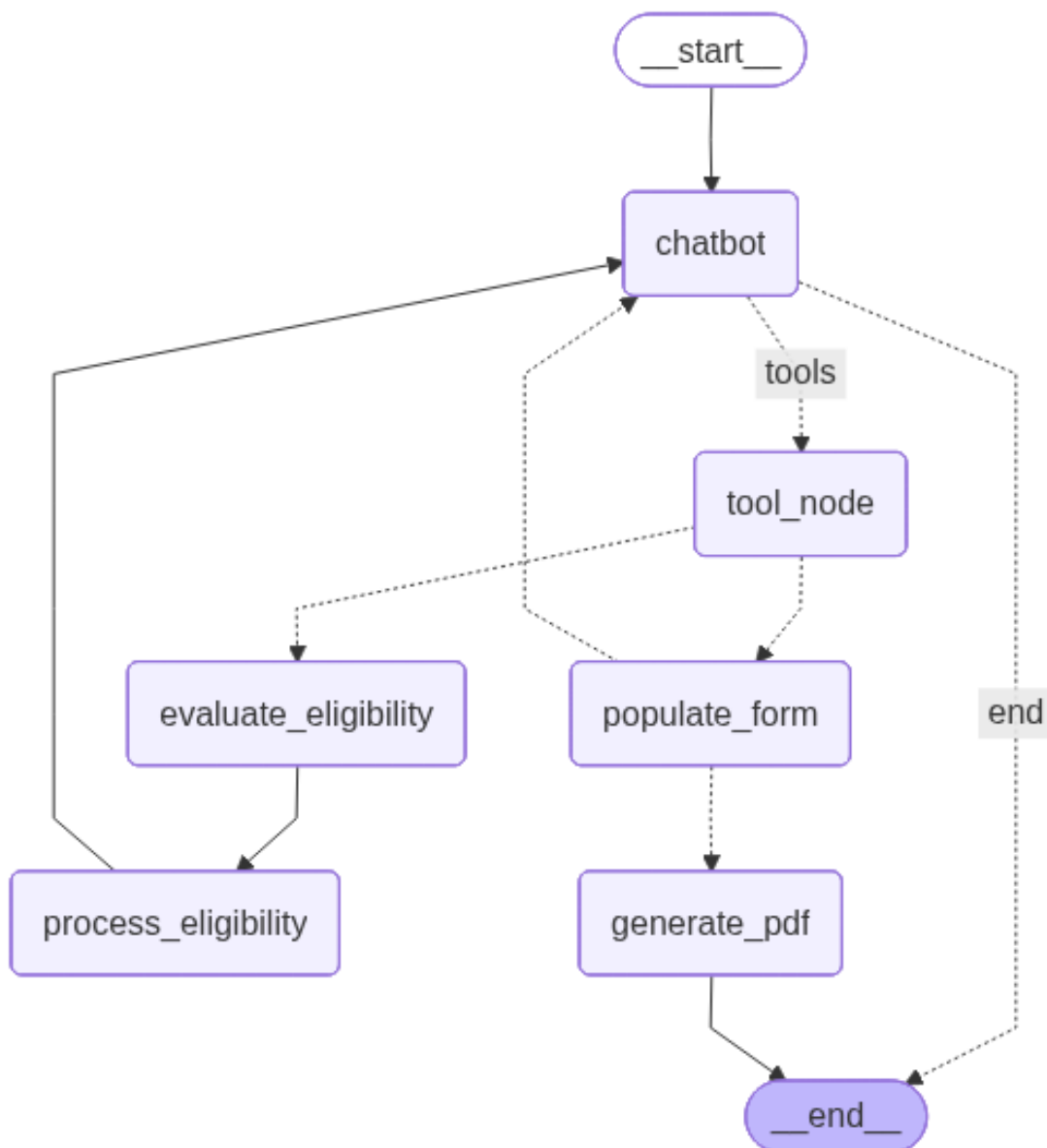


FIGURE 3.2: Flow Diagram Illustrating Multi-Agent Workflow Pipeline

The edges between nodes contain conditional logic that allows for intelligent routing through the pipeline based on state and user actions. For example, when the chatbot node detects tool calls in the conversation, it routes to the tool\_node, which then uses routing logic to determine which tool to trigger and, subsequently, which node to invoke.

The tool integration pattern utilises LangGraph's ToolNode to provide the conversational agent with specific capabilities for eligibility evaluation and form filling. The system defines two primary tools, `evaluate_eligibility_tool` and `fill_form_tool`, which are provided to the model along with descriptions of their usage. When users express intent to check eligibility, provide new information about themselves, or indicate interest in completing an eligible form, the agent should include a tool call in its response. By utilising tools, the system can leverage more specialised prompting and parsing techniques, providing better performance and more precise separation of concerns than relying on only the chatbot agent. In this case, the tools themselves are simply wrapper methods that leverage LangChain's tool integration to enable efficient node navigation. This also enables more complex conditional edges in the tool calls, providing assurances such as an eligibility check when the agent attempts to populate an ineligible form.

## 3.2 Multi-Agent Workflow Design

The system uses a multi-agent design to increase separation of concerns and distribute specialised tasks across dedicated processing nodes. Three nodes within the system contain specialised agents, each with its own aims and objectives. These are the chatbot, `evaluate_eligibility`, and `populate_form` nodes. The design follows the hierarchical delegation pattern identified in the literature, where specialised agents are optimised for specific parts of the flow while maintaining a shared state.

The chatbot node acts as the delegator and "front-end" for system interaction, guiding users through system usage and determining next steps based on user intent. This node

has the greatest level of responsibility within the system and, as such, contains the most detailed and difficult-to-construct prompt, which is available in the appendix C. The key objectives for the chatbot node are:

- conversational engagement through short, helpful messages.
- Tool call decision management based on user intent and context.
- Clear dissemination of information to the user when internal decisions are made.

The prompt for this agent was one of the longest and most error-prone parts of the system's development. As the chatbot agent must determine next moves and responses, it contains multiple objectives. When the prompt did not provide enough guidance on how to determine next moves, the agent would lean towards over-agency and attempt to respond and solve problems itself, rather than calling the internal tools. Whereas, when provided too much guidance on when to call tools, infinite loops were a risk: the agent would decide never to give an output, instead calling tools to try to determine what to do next, without any end in sight. Using the AI chatbot Amazon Q [55], this prompt was refined to strike a balance between over- and under-agency by applying rules and strategies. Still, during user testing, issues with the response style and user interface context were identified, demonstrating both risks and difficulty in guiding these agent systems.

The eligibility evaluation and populate form nodes have simpler, more focused objectives. They both produce structured outputs, which are then fed into a second processing step after the agent has performed its task. These structured outputs, integrated with the ChatOpenAI API and Pydantic typing, ensure consistent output that different parts of the flow can manage.

The eligibility assessment output consists of a list of objects, each containing 5 major fields used in processing. These are form name, eligible (boolean), missing information, needs more info (boolean), and reasoning. The following process eligibility node will

transform these into a context that can be injected into the central chatbot node. These fields are critical for the chatbot node to disseminate eligibility information to the user and determine next steps. The following is example of the structured eligibility output from the agent.

```

1      {
2          "results": {
3              "Specialised-Mobility-Aid-Request.pdf": {
4                  "eligible": false,
5                  "missing_info": [
6                      "Applicant's mobility score on a clinical 1-10 scale
7                      ↪ ",
8                      "Presence of a supporting physician willing to sign
9                      ↪ off",
10                     "Duration of established need for a mobility aid (
11                     ↪ min 6 months)",
12                     "Permanent residency status in Wales"
13                 ],
14                 "needs_more_info": true,
15                 "reasoning": "User expressed interest in wheelchair
16                 ↪ entitlement but no specific information about mobility score,
17                 ↪ physician support, duration of need, or residency was provided."
18             },
19             "...": "..."
20         }
21     }

```

FIGURE 3.3: Example JSON Output of the Agentic System's Eligibility Assessment

When `needs_more_info` is true, the chatbot node will use the missing information field to prompt the user for relevant information to determine eligibility. This is important to

ensure the chatbot stays on topic and does not ask for irrelevant or unhelpful information. When eligible is set to true and needs more info is false, the chatbot will know the user is eligible and ask whether they wish to begin filling out the form. The reasoning field is a short text extract that explains why a particular eligibility decision was made, which the chatbot node can use to describe the outcome to the user. Ultimately, the decision to move on to form-filling remains with the user, meaning they can continue to provide new information and have their eligibility re-evaluated against forms at any point.

Once a form has been deemed eligible and the user has provided intent to complete it, the populate form fields node will be invoked. The specialised agent will use a predefined field schema to extract pertinent information from the user's conversation and any uploaded documents to populate the form fields. Within the form field schema, there is a format field that guides the agent on how the text should be structured. For example, date format information such as "DD/MM/YYYY" or information on how to structure a uk national insurance number. Ultimately, despite this working well for simple fields, there is a level of over-agency, and if reworking this solution, more thorough field validation would be included through PyDantic integration.

If the agent cannot populate all fields based on the current context, it has three turns in the conversation to ask follow-up questions to fill in any missing details. The agent has been instructed to gather this information in as few messages as possible, while maintaining conversational flow. If all information cannot be gathered within three messages, the PDF generation will proceed with partial results, and the user will be warned of incomplete fields.

The culmination of this hierarchical agent model is a well-oiled machine for workflow navigation and specialised task management. Each of these agents requires specific information relevant to their task which is injected at the appropriate point. This is achieved through RAG and prompt management, which will be discussed next.

### 3.3 Knowledge Management and RAG Implementation

Knowledge management and retrieval are implemented using both Retrieval Augmented Generation (RAG) and a naive file loader. RAG is in place for eligibility retrieval to inject criteria and support form descriptions. ChromaDB [56] is the vector store of choice due to its easy integration with LangChain workflows. When building the initial Chroma vector database, OpenAIEmbeddings [57] is used as the vectorisation protocol with LangChain's RecursiveCharacterTextSplitter [58] used to chunk and process data before embedding. This workflow provides easy integration with existing tools, so no significant changes to the initial naive file retrieval were necessary. If the RAG integration fails, dependency injection ensures that the naive file retrieval can be injected in its place.

During the eligibility assessment, the user's input and conversation will be vectorised, and a similarity search will be performed against the Chroma database. The similarity search returns the most relevant eligibility criteria for their situation, ensuring that the agent is not overloaded with context it does not need. In the current implementation, however, note that with a  $K$  (number of files retrieved) of 10 and only 5 documents loaded into the database, all forms will be retrieved for every query. Ultimately, this results in a somewhat moot RAG implementation and a lack of stress testing against the benefits of RAG in this case. However, having the bones for a RAG implementation in place keeps the system extensible and scalable if load increases in the future.

Likewise, due to the sparse dataset the decision to avoid knowledge graph construction and graph retrieval was made. If many support documents in disparate languages and with varying requirements were loaded, this would be a worthwhile pursuit. However, with so little data and a lack of resources to link it, graph representation was an overhead that did not yield sufficient benefit.

The eligibility documents in the system are represented as structured Markdown files with three sections, which can be found in appendix B. These sections are the document name, description, and a list of eligibility criteria. When embedding the document into the vector database, metadata containing source file name, form name, source path, and description are provided. The eligibility criteria themselves remain in the page content, which is loaded along with the metadata as a similarity search result.

The other primary document in the system is Schema.json. This JSON file contains an object with attributes named for each form, containing metadata for the form fields within the system. The metadata contains the attributes "form" (a relative path to the actual form) and "fields", which are a list of field objects. The field objects contain a description, coordinates, format, page, and an empty text field that the populate\_form agent will fill in, and can be seen in appendix D.

Schema.json is critical for form field population, PDF annotation, and ultimately PDF generation. The decision to keep this as a structured file rather than load it into a database was made because the data is sparse. That decision works well in this case, reducing latency and dependency management, but in a scalable production application, this would need to be addressed through the use of a database.

The following section discusses how Schema.json is used for form population and the challenges with PDF manipulation that had to be overcome.

### **3.4 PDF coordinate system challenges and solutions**

Thanks to the information provided by Schema.json and its metadata, no agent is involved in the actual generation or manipulation of PDF documents. The agent's involvement in the process is limited to language extraction and transformation, keeping to its intended scope. When annotating PDFs, PyPDF is used to map to the form's specific

coordinates and to use the text output from the form-filling agent. This deterministic process ensures precise field placement and avoids the trappings and pitfalls of agent-based GUI usage and other issue-riddled solutions identified in the literature [14].

Unfortunately, despite promising results when working with some PDF viewers, consistency was a major issue. It should be noted that actual placement was reliable across viewers, but was inconsistent in precision. This was true for both bounding box placement and text placement within an annotation's bounding box. For example, the following field object, when mapped to a PyPDF TextAnnotation, produces different results when opened in Chrome vs. Mac's Preview application.

```
1 {  
2   "description": "First Name",  
3   "top_left_x": 103,  
4   "top_left_y": 663,  
5   "bottom_right_x": 289,  
6   "bottom_right_y": 639,  
7   "text": "",  
8   "format": "word",  
9   "page": 0  
10 }  
11
```

(a) Field Object Metadata



(b) Rendering in macOS Preview



(c) Rendering in Google Chrome

FIGURE 3.4: Demonstration of rendering inconsistency: (a) defines the coordinate metadata, which results in divergent visual placements between (b) macOS Preview and (c) the Chrome browser environment.

This issue was further exemplified when searching for tools that could provide precise PDF coordinate mapping. PDFs typically use a coordinate system with the origin (0,0) at the bottom-left corner of the document. However, the actual mapping of locations on the document to precise x and y coordinates usable by PyPDF as a rectangle object is not straightforward. Many PDFs contain internal metadata that shifts or transforms the

expected x and y positions, which are interpreted differently by different viewers. Many coordinate finders also place their origins at different locations based on proprietary mappings, which means the output is not usable by the system.

To add to the complexity, upon finding a working coordinate mapper, the website was taken down before form metadata could be completed, leading to another hunt for a workable tool. Ultimately, a website was found that worked with PyPDFs' coordinate mapping [59], but this highlighted a key issue with the manual process required to annotate the forms with metadata in this way. In the future, if this system is to be made scalable and workable with more forms, a solution for automatically mapping PDF form fields to coordinates must be found.

### **3.5 Session Management and Experimental Infrastructure**

Comprehensive session management capabilities are built into the system to support the experimental methodology and user study analysis. Each user interaction generates a unique session UUID, isolating the session and allowing related logs to be grouped. Quantitative measures captured include message counts, message duration, eligibility check frequency, form completion rates, and errors, providing objective performance metrics. Conversation transcripts are preserved with time stamps, enabling qualitative analysis of user interaction patterns, response quality, and conversation flow.

Each session maintains its own data directory structure, preventing cross-contamination of participant sessions while enabling the exporting of individual session data. All sessions can also be exported in a single zip file, facilitating data transfer and backup. Session deletion is password-protected, preventing bad actors from destroying research data. All session management is handled in a separate input pane that the user will not access

during the study. Outside of a user study, the project is set to private and can only be accessed by the primary researcher.

## 3.6 User Interface and Deployment Architecture

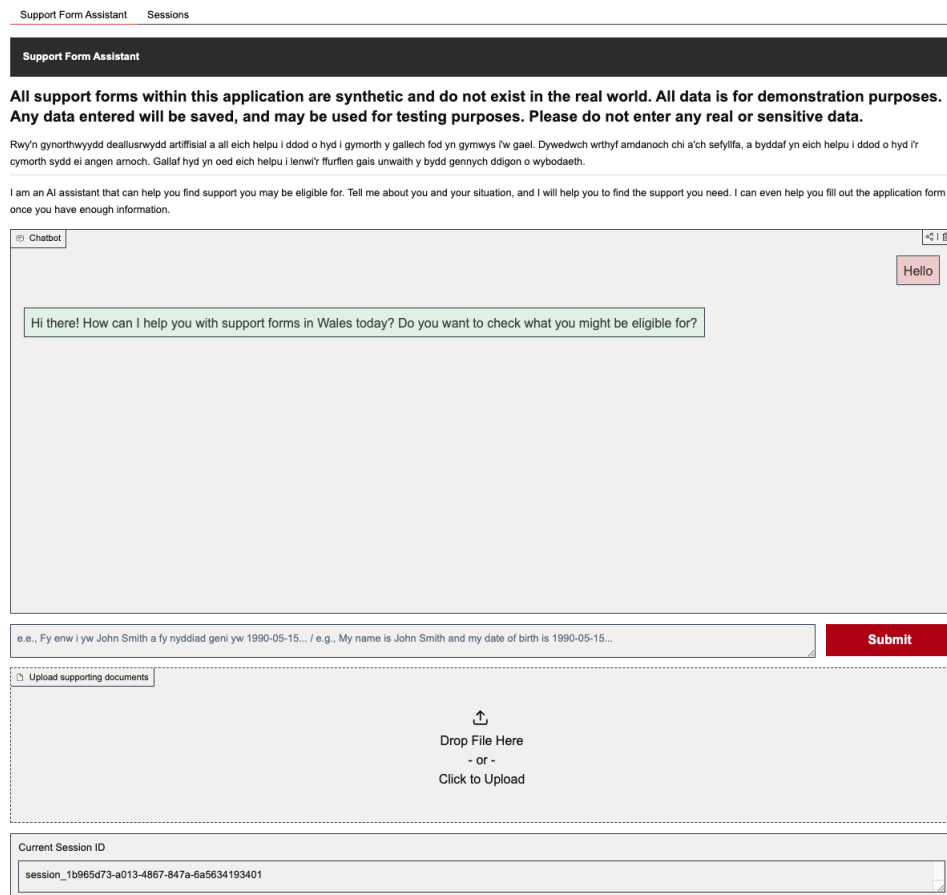
At the heart of the user experience is a Gradio-based web interface that provides an accessible, easy-to-build browser experience [60]. Gradio is optimised for spinning up chat interfaces in Python, providing useful features for agent integration and data collection. The interface uses Gradio's ChatBot component, enabling natural conversation flow with built-in message threading and visuals.

The user is presented with four main components when interacting with the system. A title and blurb serve as an introduction, setting expectations and explaining the system's use. This is presented in both Welsh and English. This is followed by a chat window, an input field, and a document upload component. After sending a message to the system via the submit button, messages will appear in the chat window, resembling chat interfaces users are likely to be familiar with. Upon generating a form, a download component will open below the chat window, and the user will be directed to access their populated form through this window.

Styling is based on the Welsh Government's [61] own design, with colours and fonts taken from their public website. This is managed via custom CSS passed to the Gradio interface for display. By staying consistent with the styling found in official sources, the system aims to increase trust through familiarity while maintaining accessibility goals set by government standards.

Deployment is managed by HuggingFace Spaces [62], leveraging cloud-based hosting to ensure each session during user testing is consistent across environments. HuggingFace offers native integration with Gradio interfaces and is the go-to hosting environment for proof-of-concept agentic solutions. All private data is accessible only via secret keys

or passwords, ensuring the deployed environment does not leak session information or sensitive logs.



The image shows a web application interface for a 'Support Form Assistant'. At the top, there are tabs for 'Support Form Assistant' and 'Sessions'. Below the tabs is a dark header bar with the text 'Support Form Assistant'. The main content area contains a disclaimer: 'All support forms within this application are synthetic and do not exist in the real world. All data is for demonstration purposes. Any data entered will be saved, and may be used for testing purposes. Please do not enter any real or sensitive data.' This is followed by a paragraph in Welsh: 'Rwy'n gynorthwyydd deallusrwydd artifisial a all eich helpu i ddod o hyd i gymorth y gallech fod yn gymwys i'w gael. Dywedwch wrthyf amdanoch chi a'ch sefyllfa, a byddaf yn eich helpu i ddod o hyd i'r cymorth sydd ei angen amoch. Gallaf hyd yn oed eich helpu i lenwi'r ffurflen gais unwaith y bydd gennych ddigon o wybodaeth.' Below this is an introductory sentence: 'I am an AI assistant that can help you find support you may be eligible for. Tell me about you and your situation, and I will help you to find the support you need. I can even help you fill out the application form once you have enough information.' The interface includes a chatbot window with a 'ChatBot' tab and a 'Hello' button. The chatbot's message is: 'Hi there! How can I help you with support forms in Wales today? Do you want to check what you might be eligible for?'. Below the chatbot is a text input field with a placeholder: 'e.g., Fy enw i yw John Smith a fy nyddiad geni yw 1990-05-15... / e.g., My name is John Smith and my date of birth is 1990-05-15...'. To the right of the input field is a red 'Submit' button. Below the input field is a section for 'Upload supporting documents' with a dashed border. It contains an upload icon, the text 'Drop File Here', '- or -', and 'Click to Upload'. At the bottom, there is a 'Current Session ID' field displaying the value 'session\_1b965d73-a013-4867-847a-6a5634193401'.

FIGURE 3.5: User Interface Design

# Chapter 4

## Methodology

The following sections will outline the experimental design and analysis plan for evaluating the agentic system.

### 4.1 Experimental Design

To evaluate the usability and viability of the agentic AI solution, a pre-post intervention study is employed based on the Technology Acceptance Model (TAM) framework [63, 64], which claims that perceived usefulness and perceived ease of use are primary determinants of technology adoption. This design allows for controlled assessment of how the system alters user perceptions and reduce barriers to support access, addressing the core research question of whether agentic AI can increase willingness to engage with support services. The pre-post design is particularly appropriate for measuring attitude changes in technology acceptance research, as it controls for individual differences while capturing the intervention effect. This approach has been validated in numerous studies examining user acceptance of novel technologies, particularly in education and public service contexts, where baseline attitudes strongly influence adoption outcomes.

The study is conducted in three distinct phases:

- Pre-intervention Phase: Participants complete an initial questionnaire to establish a baseline. This captures demographic data, general attitudes towards Welsh support services, and current temperature towards AI solutions.
- The intervention: participants are assigned a synthetic persona and supporting documents (e.g., GP note, utility bills). They engage in a role-play session, interacting with the system to determine eligibility and complete a form. During this phase, the researcher takes unstructured notes to capture qualitative queries, friction points, and user behaviours.
- Post-intervention Phase: A final survey that includes the usability questions to measure technical UX, alongside a set of control questions identical to those in the pre-intervention survey.

## 4.2 Persona and Eligibility Mapping

The study uses three personas, Tom, Sian, and Elin, carefully designed to represent varying levels of digital literacy and emotional need while ensuring validity within a Welsh support context. Each persona uses realistic demographic characteristics, support needs, and documentation patterns observed in the real world and can be found in appendix A. Persona construction mapped each character to at least one of the five synthetic forms produced explicitly for this research. This mapping provides a range of complexity across field requirements and eligibility logic, ensuring thorough system testing across different use cases. Each persona has an expected "happy path" in which the participant successfully completes a generated form, but also contains data points that trigger "unhappy paths" or ineligibility for other forms within the system. This reflects the complex reality of navigating support services.

The system contains the following synthetic forms:

- CHC Initial Screening
- Health Cost Exemption
- Local Mind Triage
- Primary Health Support Referral
- Specialised Mobility Aid Request

These forms were developed to mirror the structure and complexity of real application forms while avoiding the ethical concerns associated with them. This approach enables almost realistic testing while maintaining research ethics and participant safety. Due to limited resources, the synthetic forms are generally somewhat simplistic; however, they range in complexity across included fields and work well for testing field placement and data extraction. Each form's eligibility criteria and description can be found in the appendix [B](#).

Participants are not briefed on the criteria or available forms beforehand, so they must determine, through dialogue with the agent, whether their interaction was successful. This design choice reflects real-world conditions in which users approach support services with limited knowledge of available options.

### **4.3 Sampling Strategy and Sample Size Justification**

Sampling is conducted on a convenience basis, targeting 5-10 participants (aged 18+) from local Swansea community groups. While this represents a small sample size, it is appropriate for this exploratory research phase. The study uses a within-subjects design, in which each participant serves as their own control, thereby increasing individual statistical power compared to a between-subjects design. The research focuses on proof-of-concept evaluation and user experience insights rather than population-level

generalisation, making intensive qualitative analysis more valuable than larger statistical inference.

The sample is consistent with established practices in human-computer interaction research, where 5-8 participants often suffice to identify major usability issues and system limitations. Nielsen's research on usability testing demonstrates that most usability problems are detected with just 5 participants, supporting this sample size for identifying critical system issues and user experience patterns [65]. Sessions are conducted individually, either in person or virtually, enabling detailed observation and immediate clarification of user behaviours. This intensive approach enables rich data collection that would be impossible with larger samples, providing further depth to compensate for a lack of breadth.

## **4.4 Data Analysis Plan: Measuring the Intervention Effect**

Data collection results will produce three distinct streams:

- **Survey Data:** Quantitative Likert-scale responses (1-7) and open-ended feedback provide standardised measures, enabling pre-post comparisons and qualitative context for changes.
- **Researcher Observations:** Unstructured notes focusing on the user experience and system edge cases will capture behaviours and reactions not reflected in survey responses, helping manage potential social desirability bias in self-reported measures.
- **System Logs:** Automated logs capturing session duration, message frequency, and tool invocation counts provide objective behavioural measures independent of participant self-reporting.

The primary goal of this analysis is to compare the paired questions in both the pre- and post-intervention surveys. By calculating the delta between these scores, the research quantifies the "intervention effect". Specifically, the shift in user willingness to engage with Welsh support services, given that the system was available for production use.

Mean and median serve as descriptive statistics for analysing attitude shifts alongside usability scores. Given the small sample size, Cohen's *d* effect size calculations will be used to assess the practical significance of observed changes [66]. To mitigate the potential researcher bias inherent in small samples, quantitative results will be combined with thematic analysis of session transcripts and researcher notes. This mixed-methods approach is expected to reveal emotional themes and user experience friction points that purely qualitative measures might miss, providing a richer understanding of the intervention's impact.

# **Chapter 5**

## **Results**

This section presents the findings from the pre-post intervention study conducted with eight participants. The results are organised as follows: first, to establish the baseline characteristics and attitudes of those who took part; second, to analyse system performance; and third, to discuss the intervention effect. A mixed methods approach is used, taking into account both the qualitative and quantitative measures gathered.

### **5.1 Participant Demographics and Baseline Characteristics**

The pre-post intervention study saw eight participants successfully gather to make use of the system and complete the relevant questionnaires. Participants were between 18 and 34 years old, with the majority in the 25-34 age range. All participants were highly educated: 3 held a bachelor's degree, 4 had a master's degree, and 1 held A-levels or equivalent qualifications. The survey reported a generally high level of digital literacy, with all participants selecting either somewhat comfortable or very comfortable with technology. Most participants reported engaging with support services in the past, describing their experiences as slow, confusing, and judgmental. Of the two who had not

engaged with support services in the past, one did not know they were eligible for any support, and the other reported that the hassle was not worthwhile without a clear understanding of what they could gain.

Before using the system, several recurring barriers to seeking support were identified. The most prominent of these was the confusion around eligibility for support and the uncertainty about where to start the process. On a 7-point Likert scale (where 7 represents the strongest agreement), participants reported feeling some anxiety when faced with completing complex application forms, with a mean score of 4.875. This same average was reported when asking whether users found the official support documentation confusing ( $M = 4.875$ ); however, the range was wider, with a median of 6. This suggests that even among highly educated users, cognitive barriers remain significant.

Similarly, participants reported low confidence in investing time to seek eligibility for support ( $M = 3.375$ ) and in navigating the support application process ( $M = 4$ ). Concerns regarding accuracy were also high, with a mean score of 5.375 for the statement "I worry that I will submit incorrect information when applying for support". The initial data suggests that, even among highly literate digital users, the existing support infrastructure in Wales is perceived as confusing and intimidating.

Baseline trust in automated and AI-powered systems reported low scores ( $M = 3.375$ ), and participants are generally sceptical that AI-powered systems could accurately assess eligibility ( $M = 3.25$ ). However, there was a trend towards preferring automated systems over human advisors for simple tasks ( $M = 4.125$ ) and a willingness to engage with an AI system to help apply for support ( $M = 4.125$ ). Generally, more educated individuals are more trusting of AI systems, so this trend may indicate that the general population may see lower scores here [46, 47]. This data, however, is too sparse to confirm that sentiment.

## 5.2 System Performance and Technical Metrics

The system's technical efficacy was evaluated using session logs that captured multi-agent orchestration, tool usage, and interaction duration. Data was collected from the eight study sessions to determine how effectively the autonomous workflow handled real-time eligibility screening and document manipulation. Across these sessions, the system managed a total of 148 messages, with an average of 18.5 exchanged per user.

Efficiency was measured by session duration, the frequency of tool calls, and the number of forms the user downloaded. The mean session length was 9.4 minutes, and the eligibility agent was invoked 2-3 times per session. This suggests that the workflow successfully utilised an iterative feedback loop to evaluate eligibility as information was made known to the system.

The form filler agent and PDF manipulation tools demonstrated high reliability in structured data extraction. Of the eight participants, seven reached the form generation stage, meaning at least one successful outcome for all available forms and personas. A total of 8 successful form generations were recorded across all sessions, with one user generating two forms at different times following a re-evaluation of their needs. All generated forms were successfully and precisely populated with information parsed from user input, demonstrating that the form-population pipeline was highly effective. The first session identified an error in the population of one of the form fields, which caused fields beyond the first page to remain unfilled. This was due to human error and was resolved before subsequent participation to great success.

Despite this success, the researcher notes highlighted some technical friction regarding the agent's awareness of its own interface. In multiple sessions, the chatbot incorrectly told users they could not upload documents directly, leading users to copy and paste text instead. One user attempted a drag-and-drop upload that only worked once, leading to confusion when subsequent uploads failed. The logs indicate that while the system is

stable, it occasionally hallucinates during the form-filling process. In one session, the bot erroneously checked "Yes" for a low-income scheme that the user had never mentioned.

The system reported high stability throughout the intervention with no reported errors or crashes. This indicates that the multi-agent handover, workflow, and RAG lookups were working well. The logs further indicate that the system's agentic nature and its ability to remember state across messages were a significant boon. For example, in one session, the system correctly identified that a form was ineligible because the GP referral was outdated. Subsequently, it pivoted the conversation to explain how the user could obtain a new referral. This demonstrates the system's ability to go beyond a linear chatbot, providing a reasoning engine that helps users understand the logic behind support criteria directly relating to their circumstances.

### **5.3 Impact on Barriers to Support**

Comparison of pre- and post-intervention survey data shows a clear shift in perceptions of eligibility, confidence, and willingness to engage. The most significant change was towards willingness to invest time in seeking information when unsure of eligibility. Before using the system, users reported a mean score of 3.375, indicating that they believed the perceived hassle outweighed the benefits. Following the interaction, this mean score increased significantly to 5.25, suggesting that the system successfully reduced the perceived cost of information seeking.

| Paired Question Statement                                                                                                 | Pre-Mean | Post-Mean |
|---------------------------------------------------------------------------------------------------------------------------|----------|-----------|
| If I knew I were eligible, I would be willing to apply for support from the Welsh government or a charitable organisation | 5.875    | 6.125     |
| I believe that support services are relevant to people in my situation                                                    | 4.625    | 4.75      |
| I feel confident about successfully navigating the process of applying for support                                        | 4.00     | 5.375     |
| Even if I were unsure of my eligibility, I would still invest time in seeking information about support available to me   | 3.375    | 5.25      |
| I would recommend applying for support services to someone in a similar situation to mine                                 | 4.50     | 5.00      |

TABLE 5.1: Comparison of Means for Paired Pre- and Post-Intervention Questions

The overall intervention effect across all paired survey questions yielded a Cohen's  $d$  of 1.02, indicating a large effect size according to conventional interpretations (small = 0.2, medium = 0.5, large = 0.8) [66]. This effect size suggests that the system intervention produced meaningful changes in users' attitudes towards engaging with support services. Confidence in the application process also improved, rising from a mean score of 4 to 5.375. This shift aligns with the system's technical success, which consistently performed well in data extraction and field mapping during user sessions. Participants also reported general satisfaction with the system's ability to determine eligibility, suggesting that a reduction in cognitive load may have contributed to the shift.

General sentiment regarding relevance and recommendation also trended upwards. The belief that support services were relevant increased from 4.625 to 4.75, while the likelihood of recommending applying for support to others rose from 4.5 to 5. The baseline willingness to apply for support if eligibility were confirmed rose from 5.875 to 6.125

post-intervention. These figures indicate that, while participants were already predisposed to seeking help, the system could serve as a catalyst by making it easier to get started.

The large effect size ( $d = 1.02$ ) provides evidence that the observed changes represent genuine intervention effects rather than measurement error or random variation. In the context of technology acceptance research, effect sizes above 0.8 are considered indicative of meaningful practical impact [66], suggesting that the system successfully addressed key psychological barriers to support service engagement. This finding is particularly significant given the small sample size, as large effect sizes indicate that the intervention impact was sufficiently robust to be detected despite limited statistical power [65]. Given the limited sample size, it is pertinent that this intervention effect not be generalised beyond the demographics sampled here.

| Participant ID                | Pre-Mean    | Post-Mean |
|-------------------------------|-------------|-----------|
| Participant 1                 | 4.40        | 5.40      |
| Participant 2                 | 4.20        | 7.00      |
| Participant 3                 | 4.40        | 4.20      |
| Participant 4                 | 4.00        | 3.80      |
| Participant 5                 | 5.40        | 6.40      |
| Participant 6                 | 4.20        | 4.80      |
| Participant 7                 | 4.40        | 4.60      |
| Participant 8                 | 4.80        | 6.20      |
| Mean                          | 4.475       | 5.3       |
| Standard Deviation (SD)       | 0.412       | 1.07      |
| Pooled SD                     | 0.81        |           |
| <b>Cohen's <math>d</math></b> | <b>1.02</b> |           |

TABLE 5.2: Individual Participant Means and Effect Size for Paired Questions

## 5.4 User Feedback

Qualitative data about the interaction with the system was gathered through two open-ended questions following the intervention, which you can find a compilation of responses to in the appendix [E](#):

- What did you find to be the most helpful feature or aspect of the system?

- What was the most frustrating part of using the system, and how could it be improved?

Participants consistently noted the system's speed in both latency/response times and its ability to speed up information gathering. This reinforces the decision to use smaller LLM models, specifically OpenAI's 4o-mini, rather than their newer, larger models, due to 4o-mini's snappy performance. Not only are the smaller models more responsive, but they can also deliver more succinct, on-topic messaging because their training sets are smaller and more focused. During development of the system, OpenAI's GPT-5 model was both significantly slower and more likely to stray from the intended goal of its prompt. This could be related to the different prompting strategies required for naive language models compared to newer reasoning models, but it was an interesting observation.

Another helpful feature noted was the system's ability to deduce and retrieve information during the interaction, without the participant needing to state it explicitly. For example, one participant attempted to scan their persona's supporting documents for specific medical information but was unable to parse them efficiently. Upon uploading the documents to the system, the agent was aware of the required information and could complete the checks correctly. This could significantly reduce the burden on users, allowing them to focus on the information that matters, rather than the minutiae and technicalities. There is also potential for this to be an accessibility bonus, as those with difficulty reading or parsing this information can rely on the system to do this for them.

Unfortunately, the system's ability to understand and provide information was also noted as a frustration. One participant noted that the system ticked an incorrect answer in the form, without asking for clarification. Incorrect answers within an application document, especially when the system marked 'yes' for whether the user was applying for a low-income scheme, are a serious concern for such a system. Despite notices when the system generates a document for the user to review the answers; likely, many would

accept and submit whatever the system offers. Even among this highly educated and AI-sceptical group, responses to "I would be comfortable submitting an application form generated by the system" had a mean of 5.125. Although conjecture in this case, understanding user habits and the likelihood of user reviews of produced documents is critical to assessing the security risk a flaw like this poses, weighed against the benefits of the system.

Another major frustration among participants, noted both in the survey and by the researcher, was the agent's UI blindness. In multiple instances, the user would ask the system whether they could, or were allowed to, upload their supporting documents. This was surprising, as the "upload supporting documents" component was present and the participants were provided time to familiarise themselves with the interface. Unfortunately, as the agent had not been prompted with specific knowledge of the user interface, it stated in each instance that it could not accept document uploads. This led to an inefficient exchange, in which participants attempted to extract the information they needed from the documents so it could be entered as text.

The apparent need for guidance through the process and the transfer of some user agency to the system were both concerning and unexpected. The concern arose because this was a highly digitally literate cohort who, in other cases, could likely have overcome any UX limitations to understand the system's capabilities. In this case, the presence of an "intelligence" seems to have dampened this willingness to experiment with the system and test its capabilities. For a cohort that requires greater accessibility support, this could be truly destructive to the experience and lead to unnecessary digital exclusion.

In the future, this instance can be fixed with more robust prompting or other methods to provide interface context to the chatbot agent. It does, however, still highlight the risks of these "reasoning" agents, as although the agent specifying that it cannot accept uploads appears reasonable, it can not and did not know that this was the case. The production

---

of confident, reasonable falsehoods, especially among those who have come to a system seeking guidance, is an extreme security concern that should not be overlooked.

# Chapter 6

## Discussion

The study's findings demonstrate a successful shift in user attitudes, with a large effect size (Cohen's  $d = 1.02$ ) indicating that agentic AI can meaningfully increase willingness to engage with support services. However, combining the experimental data with additional user feedback makes it clear that there is more to consider than this one successful intervention. Technical efficiency must be balanced against reality and the more diverse psychological needs of the Welsh population.

This discussion will focus on the broader implications of this study. During the post-intervention survey, participants were asked to share "any additional thoughts about how this system might impact access to support services in Wales", providing further topics for discussion.

### 6.1 Lowering Barriers to Support

The results showed a meaningful increase in willingness to seek support regardless of eligibility, rising from a mean of 3.375 to 5.25. As mentioned previously, reduced cognitive load during the administrative process is likely to have contributed to this increase. There may, though, be a more profound psychological benefit to an agentic system in

this space, namely a depersonalisation of the experience. A participant noted that for those with social anxiety, interacting with an AI is "more comfortable" than speaking to a human. This suggests that the system saves time while also providing a safer starting point for understanding how support may apply to individuals. This is especially true for those who may otherwise experience social anxiety or are too nervous to engage with human-facing services.

The system also demonstrated an ability to bypass jargon, potentially addressing an accessibility gap. Participants highlighted that the AI could help those with "difficulties in processing information" access services without needing to understand the complex language of official forms. The system has the potential to serve as a bridge to digital inclusion, but this must be balanced with the noted difficulties of agentic LLMs in general. Can we trust an agent's output? Or are we breaking down barriers to replace them with even more sinister issues?

## **6.2 The "Wait List" Paradox and Systemic Impact**

An interesting piece of feedback from one participant was that this system does not address the true crisis in support-seeking and may even cause further harm. The participant noted that "waiting lists... are still the main issue" and that "this could cause a backlog... extending the wait list". The idea that a finalised version of such a system could make applying for support services too efficient, inadvertently placing further strain on the already overstretched Welsh NHS, warrants thought. Placement of such a system within a triage pipeline, where a user gains access after being assessed as potentially benefiting, could reduce the burden in this case but would nullify some of the system's benefit. Future iterations must consider how AI can be integrated into the entire service pipeline, with analysis not only of its individual benefits but also of any systemic harms that could result.

### **6.3 Technical Friction as a Barrier to Trust**

The UI blindness observed in the study, while a naive bug that should have been identified earlier, represents a broader issue with agentic systems. Confidently wrong assertions within such systems, while a frustration for digitally literate populations, can be truly destructive to trust in the system by broader populations. Agentic systems are increasingly perceived as an "authority" in interaction, as demonstrated in this small study. Technical hallucinations about capabilities can lead to real digital exclusion, and even if the system prompts were more robust, drawn-out interactions can still lead to context rot and similar issues. Unfortunately, for those vulnerable populations dealing with anxiety or who are generally less digitally capable, bugs like these can lead to abandoning the process entirely after placing the blame on themselves for lack of understanding. This reinforces the need for the 5Ps of ethical data preparation, highlighting the need for the system to not only be accurate in its logic but also transparent about its own limitations.

### **6.4 Autonomy vs Safety**

The most concerning technical finding was the system's tendency to hallucinate information, such as erroneously checking "Yes" for applying to a low-income scheme. While the group had a high mean of 5.125 for comfort in submitting the applications produced by the system, the group was highly educated and likely to review the forms before submission. In reality, at least anecdotally, many would probably skip the verification step and accept the system's output.

User feedback explicitly called for a "clarification step" before documents were finalised and generated by the system. This suggests that total automation, as the system is currently, is inappropriate for a system in this space. Instead, a human-in-the-loop model should be put in place, where the AI's reasoning engine provides a draft of answers with

justifications before generating a document. This was not implemented during development to reduce cognitive load and friction, but it is now clear that some level of human verification is critical for such systems. A verification step could also mitigate the risks of black-box decision-making and produce greater trust in the agent's output.

## **6.5 Conclusion to Discussion**

While the agentic system effectively reduces the administrative and psychological hurdles to seeking help, there are major considerations that must be addressed before it can be deployed in the Welsh public sector. The system's success depends not only on its technical merit but on its ability to reduce the burden on individuals and the system, ensuring that an increased burden is not placed on existing infrastructure due to errors and an influx of applications.

# Chapter 7

## Limitations

This research faces several limitations that constrain generalising the findings to broader populations. A substantial limitation is the narrow demographic composition of the participant sample. All eight participants were highly educated and digitally literate, aged 18-34, and had at least some exposure to autonomous systems. This represents a population segment that is likely more tolerant of system imperfections and better equipped to identify potential errors. The demographic homogeneity is also a problem, as support services typically serve a diverse population, including older adults, those with lower levels of education, and those experiencing digital exclusion. The literature suggests that trust in AI systems varies significantly with educational background, with higher education correlating with higher AI acceptance. This means the positive results observed may not translate to the populations most in need of support services, potentially leading to an overestimation of the system's real-world effectiveness.

The use of synthetic forms and personas, which are necessary for ethical reasons, introduces another layer that may limit the applicability of the findings. Real Welsh government forms are likely more complex, with intricate dependencies and legal language that could challenge the system's capabilities beyond what was observed. Likewise, the

controlled nature of the synthetic personas may not capture the complexity that characterises real support situations. In these situations, individuals may be under stress and less able to verbalise their needs or identify system errors.

The coordinate-based PDF manipulation approach, while innovative in this case, represents a significant scalability limitation. The manual annotation process required for each form would be prohibitively expensive for large-scale deployment across the diverse range of Welsh government forms. This approach also assumes static form designs, which may not reflect the reality of evolving government documentation. Many modern application forms don't even use the PDF format, limiting the system's usefulness in many scenarios.

The sample size ( $n = 8$ ) limits statistical power and prevents meaningful processing and subgroup analysis. While the pre-post design controls for individual differences, the lack of a control group means that observed changes cannot be definitively attributed to the system intervention. They could instead be attributed to uncontrolled variables, such as increased awareness of support services or to research bias leading to improved scores.

Finally, the study's focus on immediate post-intervention effects provides no insight into longer-term user behaviour or system reliability. The sustainability of observed attitude changes and the system's performance over extended use periods remains unknown, representing a critical gap for understanding real-world viability.

# Chapter 8

## Future Work

The findings of this study provide a foundation for several avenues of technical refinement and research. While the agentic workflow demonstrated a capacity to reduce the administrative burden of seeking support, the transition from a proof-of-concept to a production-ready government tool requires addressing several challenges.

### 8.1 Reliability and Verification

Future iterations must move beyond the current over-agency present in this study. Production systems in high-stakes environments can not be allowed to autonomously complete form fields without explicit verification. This presents opportunities for multiple solutions.

An antagonistic verifier agent could be put in place to provide internal verification on output. The verifier agent would check all output against the source context to ensure that all statements and filled fields are valid and correct. To prevent further hallucinations from the verifier, multiple verifiers could run in parallel, with a quorum required to validate the output and flag hallucinations.

A human-in-the-loop integration should also be considered to ensure that the user retains control of their information and provides explicit consent for data entry. To reduce the burden on users, this could be implemented on more complex or high-stakes fields, rather than requiring explicit verification at every stage. In either case, the user should be involved to ensure trust is maintained and data integrity remains core.

## **8.2 Scalability**

The current coordinate-based PDF manipulation offers a deterministic alternative to unreliable GUI methods but currently relies on a manual human labelling step to map field coordinates. In a production system, this would cause significant scalability concerns and bottleneck the process. Instead, computer vision could be explored to find and map form field coordinates for the PyPDF system. The storage and retrieval of these coordinates during runtime should also be enhanced by moving them into a relational database for rapid, scalable retrieval.

## **8.3 Knowledge Representation and GraphRAG**

While the current RAG implementation is in place, the circular conversations and context misses noted during the study session suggest a limitation in the system's multi-hop reasoning. In a production environment with greater resources, more data, and greater complexity, KG-enhanced RAG would enable the agent to map dependencies among support services more effectively. For example, linking employment status or income to financial support could help proactively select or rule out entire categories of support based on fewer attributes.

## **8.4 Long Term Impact and Digital Exclusion**

The current study was constrained by a small, homogenous sample ( $N = 8$ ) and focused on immediate post-intervention effects. Future research should track behaviour over a longer period to assess whether the increased willingness to engage leads to actual successful outcomes. Targeted studies with more vulnerable populations, such as older adults and those with lower literacy, are essential to ensure that the interface is accessible for the true target demographic.

## **8.5 Systemic Issues**

Finally, research is required to understand the systemic impact of such a system on support services and the NHS itself. Future work should investigate how agentic systems can be integrated into the triage pipeline to ensure backlogs are manageable and that solving one problem with AI does not create more in human-led services.

# Chapter 9

## Conclusion

This research aimed to answer whether an agentic AI system could increase willingness to engage with support services in Wales. Based on the findings, it would be too easy to say yes, and simply leave it at that. There are critical caveats regarding safety and human oversight that must be addressed before such a conclusion could be made. However, the implementation of a multi-agent architecture using LangGraph and a novel coordinate-based PDF manipulation tool demonstrates that there are avenues to success in the public sector through AI.

The intervention effect, specifically a Cohen's  $d$  of 1.02, demonstrates a large effect size in shifting user attitudes towards support seeking from a hassle to a manageable process. While the literature often highlights the potential for AI for personalisation, this research provides concrete evidence that agentic reasoning can lower psychological barriers for users, even among the AI sceptical. The technical success was, however, tempered by the discovery of hallucination risks in autonomous form filling, which confirms that full autonomy is currently inappropriate for high stakes public service domains.

While agentic AI offers a path for digital inclusion in Wales, its deployment must strike a difficult balance between autonomy and human involvement. However, the framework established here provides a promising blueprint for future development. Future work

---

must now focus on the systemic impacts to support infrastructure, and the removal of critical risks identified.

# Appendix A

## User Personas

The following sections highlight the three personas used during the user study and the information provided to the participants relating to them.

### A.1 Elin

Your Information Name: Elin Quin Email: elin@example.com Date of Birth: 21/03/1990 (35 years old) Address: 15 Heol Dewi Sant, Llanelli, Wales, SA15 3YB Your Situation Condition: Severe anxiety and depression Symptoms: Social Anxiety, constant worry, insomnia, can't leave the house most days Severity: Feeling overwhelmed and desperate Your history - Resistant to drawn out interactions - Overwhelmed by phone calls and paper forms - Want quick access to help - Anxious about the process Goal You want counselling or therapy. You aren't sure what you're eligible for, but you don't want to drag this out. Supporting Docs 1. Referral letter from doctor dated three months prior 2. An email thread with Mind 3. A personal statement on current condition

## A.2 Sian Stewart

Your Information Name: Sian Stewart Date of Birth: 14/05/1962 (63 years old) NHS number: H6789012A Address: 6 Ferrara Quay, Maritime Quarter, Swansea, SA1 1UQ  
Your Situation A single parent whose hours have been cut. Annual income: £13,500 Savings: £5000 Health: You suffer with COPD (Chronic obstructive pulmonary disease) and were recently discharged from hospital after acute exacerbation Mental Health: Ongoing financial stress and health worry Your Goal You are trying to find any financial support you may be eligible for. You are busy and cannot be bothered to check dig for extra paperwork, so you've collected what was on hand. You're worried your savings may disqualify you from support. You are mindful that you could benefit from mental health support too. Supporting Docs 1. A bank statement showing savings balance of £5000 2. A payslip showing reduced income of £13,500 annually 3. Hospital discharge letter confirming chronic COPD 4. Welsh water utility bill containing address

## A.3 Tom Davies/ Michael Davies

Your role You are Tom Davies, full-time unpaid carer for your elderly uncle, Michael Davies. Michael's Information Name: Michael Davies Date of Birth: 12/03/1940 (85 years old) NHS Number: PE123408A Residential Status: Own home (you live in and provide care) Address: 72 Ayelsbury Road, Swansea, SA1 1PQ Michael's Care Needs Mobility: Requires full hoisting, two people for transfers Communication: Severely impaired Psychology: High anxiety, needs hourly check ins Aids in use: Specialist chair, hoisting equipment Your Goal You are currently privately funded and in desperate need of financial support. You are hoping to get a new wheelchair for your uncle, but you aren't sure if you are eligible or what routes exist. Supporting Docs 1. Physio report documenting mobility score 2. Recent care notes 3. Michael's discharge care plan 4. A GP note confirming Michael is a patient and that Tom is his carer

# Appendix B

## Support Forms and Eligibility Criteria

This section will demonstrate the criteria used by the system to determine user eligibility in it's raw markdown form prior to parsing and storage in the vector database.

### B.1 CHC Initial Screening

# CHC Initial Screening

## Description Initial screening for Continuing Health Care funding to cover ongoing health and care needs for individuals requiring substantial support.

# Eligibility - Applicant must have a severe limitation in mobility, or a severe limitation in communication (requiring specialist aids). - Applicant must exhibit high psychological or emotional needs (e.g., requiring hourly checks or constant supervision). - The applicant must not be currently receiving long-term care funded by a local authority Social Services department. - All of the above must be true.

## **B.2 Health Cost Exemption**

### **# Health Cost Exemption**

**## Description** Exemption from NHS prescription charges, dental costs, and optical expenses for individuals on low income with chronic health conditions.

**# Eligibility** - Applicant must have an annual net household income of less than £14,000.  
- Applicant must have combined savings or investments of £16,000 or less. - Applicant must be either a full-time student, or must have a chronic medical condition (e.g, Diabetes, COPD, etc.) - All of the above must be true.

## **B.3 Local Mind Triage Request**

### **# Local Mind Triage Request**

**## Description** Access to mental health support services through Local Mind, including initial assessment and signposting to appropriate mental health resources.

**# Eligibility** - Applicant must be 18 or older. - Applicant must be a current resident of Wales. - Applicant must have not accessed this specific service within the last 6 months.  
- All of the above must be true.

## **B.4 Primary Health Support Referral**

### **# Primary Health Support Referral**

**## Description** Referral for psychological support services including counselling, therapy, peer support groups, or mindfulness training through NHS Wales.

# Eligibility - Applicant must be seeking support via "Counselling/ Therapy" OR "Peer Support Group" OR "Mindfulness Training". Use best judgement to determine intent in line with one of these options. - Applicant must have a current (dated within the last 3 months) GP referral. - Applicant must be registered with a Welsh NHS GP Practice. - The stated needs must be psychological in nature (e.g., anxiety, depression, trauma). - All of the Above must be true.

## **B.5 Specialised Mobility Aid Request**

# Specialised Mobility Aid Request

## Description Request for specialised mobility equipment such as wheelchairs, walkers, or mobility scooters for individuals with significant mobility limitations.

# Eligibility - Applicant's mobility score must be 3 or less on a clinical 1-10 mobility scale (1 being worst). - Applicant must have a supporting physician willing to sign off on the request. - Applicant must have an established need for the a mobility aid for at least 6 months. - The applicant must be a permanent resident of Wales. - All of the above must be true.

# Appendix C

## Agent Prompts

This appendix contains the complete logic for the three primary agents. These prompts are dynamically injected with state data at runtime to enable context-aware behaviour.

### C.1 Form Filling Agent

The following prompt dictates how the agent should manage information extraction for filling form fields. For example, instructions on what to do if information is missing.

```
1 def get_form_filling_system_message() -> str:
2     datetime = get_current_datetime_context()
3
4     return f"""{datetime}
5     You are a helpful assistant that extracts information and
6     fills form fields accurately.
7
8     CRITICAL INSTRUCTIONS:
9     1. Extract information from the ENTIRE conversation,
10    with special attention to the MOST RECENT messages
```

2. If a field is marked as MISSING, look carefully in the latest user messages for that information

3. If a field is already filled, preserve that value unless new information is provided

4. ONLY output the JSON structure requested - no additional text

5. CRITICAL: Each field must contain ONLY information relevant to that specific field's description

6. NEVER combine answers to different questions into a single field

7. If the user answers multiple questions in sequence, match each answer to its corresponding field ONLY

When filling fields:

- Use exact text from the conversation when possible

- For dates/numbers, format according to the field format specification

- For multiple choice, use the exact option text provided

- Leave fields empty only if no relevant information exists in the conversation

ABSOLUTELY FORBIDDEN:

- NEVER put explanatory text like "not provided", "missing", "unknown" in the text field

- NEVER put phrases like "is not mentioned" or "cannot find" in the text field

- NEVER put answers to OTHER fields into the current field (e.g., don't put "No" from a Yes/No question into an NHS number field)

```
26     - NEVER concatenate multiple pieces of information that
    belong to different fields
27     - If information is missing, leave the text field as an
    EMPTY STRING ""
28     - The text field must contain ONLY the actual data
    value for THAT SPECIFIC field, or be empty
29     """
```

LISTING C.1: Form Filler System Message

The following prompt provides the full conversation and document context to the agent, as well as information on fields that have already been completed so that superfluous responses are not returned.

```
1 def get_form_filling_prompt_template(
2     form_name: str,
3     field_descriptions: str,
4     already_filled_section: str,
5     multiple_choice_instruction: str,
6     user_text: str,
7     document_section: str,
8     schema_form: str
9 ) -> str:
10     """Template for form field population prompt."""
11     return f"""You are an assistant that fills out form
    fields based on user-provided information.
12
13     Form: {form_name}
14
15     Fields to fill:
16     {field_descriptions}
17     {already_filled_section}
```

```
18     {multiple_choice_instruction}
19     User-provided information (extracted from conversation)
20     :
21     {user_text}
22
23     Additional context extracted from uploaded documents:
24     {document_section}
25
26     Instructions:
27     - Extract the relevant information from the user's text
28       for each field.
29     - Return a JSON object with the exact same structure as
30       the input schema.
31     - Fill in the "text" field for each field entry.
32     - Keep all other fields (description, coordinates,
33       format) exactly as they are.
34     - PRESERVE existing values for fields that are already
35       filled (marked as CURRENTLY FILLED above).
36     - Only update a field if:
37       a) It is currently empty, OR
38       b) The user explicitly provides new information for
39         that specific field
40     - If information is missing for a field and it's not
41       already filled, leave "text" as an empty string "".
42     - Respect the format field for each form field and
43       ensure output text matches that format.
44     - For multiple choice fields: Use the EXACT option text
45       from the provided list, not variations.
46
47     Return ONLY valid JSON matching this structure:
```

```
39     {{
40         "form": "{schema_form}",
41         "fields": [
42             {{
43                 "description": "...",
44                 "top_left_x": ...,
45                 "top_left_y": ...,
46                 "bottom_right_x": ...,
47                 "bottom_right_y": ...,
48                 "text": "...",
49                 "format": "...",
50                 "page": ...,
51                 "choices": [...]
52             }}
53         ]
54     }}
55     """
```

LISTING C.2: Form Filler Prompt

## C.2 Eligibility Evaluation Agent

This prompt dictates how the agent should reason with the eligibility criteria. `DateTime` is injected to ensure any temporally restricted eligibility criteria (referral must not be x months old etc) can be correctly assessed.

```
1 def get_eligibility_evaluation_system_message() -> str:
2     datetime = get_current_datetime_context()
3     return f"""{datetime}
```

You are an eligibility assessment assistant. Evaluate the user's eligibility for each available support form.

For each form, assess:

1. eligible: boolean - whether the user appears eligible based on available information
2. missing\_info: list[str] - information needed to make a more confident assessment
3. needs\_more\_info: boolean - whether additional information is required to determine eligibility
4. reasoning: str - brief explanation of the assessment

#### CRITICAL EVALUATION PRINCIPLES:

**\*\*For OR conditions (e.g., "must be student OR have chronic condition"):**

- If ANY option in the OR is satisfied, the entire condition is MET
- Do NOT include other options from the same OR condition in missing\_info
- Only include an OR condition in missing\_info if ALL options are unknown

**\*\*For missing\_info field:**

- Include ONLY information that is genuinely unknown AND required for determination
- Do NOT include alternatives from satisfied OR conditions
- Use ALL available information from conversation AND documents

```
23     - Be specific about what information is needed
24
25     **Decision Logic:**
26     - Set eligible=TRUE when all criteria are satisfied
27     - Set eligible=FALSE when any hard requirement is
    definitively not met
28     - Set needs_more_info=TRUE when missing information
    prevents definitive determination
29     - Set needs_more_info=FALSE when you can make a
    definitive eligible/ineligible decision
30
31     **Hard Requirements vs Missing Information:**
32     - If most criteria are met but only missing 1-2 pieces
    of information, consider needs_more_info=TRUE
33     - If fundamental requirements are not met, set eligible
    =FALSE and needs_more_info=FALSE
34
35     IMPORTANT: Avoid contradictions between reasoning and
    missing_info. If reasoning states a condition is satisfied,
    do not list it as missing.
36     """
```

LISTING C.3: Eligibility Evaluation System Message

This prompt provides the full context to the agent.

```
1     def get_eligibility_evaluation_prompt(
2         eligibility_summary: str,
3         user_conversation: str,
4         document_context: str
5     ) -> str:
```

```
6         return f"""Evaluate the user's eligibility for each
7         available support form based on the following information:
8
9         {eligibility_summary}
10
11        User Information (from ENTIRE conversation history):
12        {user_conversation}
13
14        Additional Context from Uploaded Documents:
15        {document_context}
16    """
```

LISTING C.4: Eligibility Evaluation Prompt

### C.3 The Orchestrator (Main Chatbot Agent)

This prompt is the most complex as it dictates how the agent should act in different conversation "phases" as well as instruction on when to use tool calls.

```
1 def get_chatbot_prompt(state: ChatState) -> str:
2     context = build_context(state)
3     datetime = get_current_datetime_context()
4
5     return f"""{datetime}
6
7     You are a warm, conversational support form assistant
8     helping users in Wales find and fill support forms.
9
10    {context}
11
12    ## Core Workflow
```

```
11
12     1. **Eligibility Check**: When user shares their
situation -> use `evaluate_eligibility_tool`
13     2. **Form Selection**: Present eligible forms -> wait
for user to choose
14     3. **Form Filling**: User selects form -> use `
fill_form_tool(form_name="...")`
15     4. **Missing Fields**: If fields missing -> ask
conversationally for ALL at once -> user provides info ->
call `fill_form_tool` again
16
17     ## Response Strategy
18
19     **When user requests help and eligibility results show
:**
20     - Forms marked as eligible -> Present them for
selection
21     - Forms with needs_more_info=TRUE -> Proactively ask
for the specific missing information if the form is relevant
to the user and they have not yet selected an eligible form
22     - **IMPORTANT: For forms marked "Not eligible", explain
why they don't qualify**
23     - Focus on what IS available while being transparent
about what isn't
24
25     **Communication Principles**
26     - Be transparent about ineligibility reasons - users
need to understand why forms aren't available
27     - When explaining ineligibility, be empathetic but
clear
```

```
28         - After explaining ineligibility, redirect to available
options
29         - Don't leave users wondering why relevant forms aren't
mentioned
30
31         **Information Gathering Principles:**
32         - When forms need only 1-2 pieces of missing
information, ask for those specific items
33         - Focus on forms most relevant to the user's stated
needs
34         - Ask for missing information conversationally and
specifically
35         - Group related missing information together in one
request
36
37         **Prioritization Logic:**
38         - If user requests specific type of help (financial,
health, etc.), prioritize forms that address that need
39         - If multiple forms need information, start with the
one requiring the least additional information
40         - If a form is very close to being eligible, mention it
specifically
41
42         ## Tool Usage Rules
43
44         **evaluate_eligibility_tool:**
45         - CRITICAL: Use when user provides ANY information that
could affect eligibility (age, residency, income, medical
conditions, etc.)
46         - Use when: User asks "what am I eligible for?"
```

```
47         - Use when: User answers questions about eligibility
criteria
48         - NEVER assume eligibility yourself - ALWAYS call this
tool to verify
49         - DON'T use when: Already filling a form and user only
providing missing field data
50
51     **fill_form_tool:**
52         - ONLY use when: Form is in the **Eligible** list above
AND user explicitly selects it
53         - Use when: User provides missing field information (
and current_form_name is set)
54         - NEVER use when: Form is not in **Eligible** list (
call evaluate_eligibility_tool first)
55         - DON'T use when: In FORM SELECTION PHASE (present
forms first)
56         - DON'T use when: Missing fields listed above (ask
conversationally instead)
57         - DON'T use when: Just called it (wait for result)
58         - IMMEDIATELY use when: User selects any form from the
**Eligible** list above
59
60     ## Response Style
61
62         - Keep responses SHORT and conversational (2-3
sentences max)
63         - Ask for multiple missing fields in ONE message,
grouped naturally
64         - Say "may be eligible" not "are eligible" until
evaluate_eligibility_tool confirms
```

```
65         - Don't over-explain - be direct and helpful
66         - If uncertain forms exist, mention them briefly and
ask for missing info
67         - Use information already provided from conversation
AND documents
68
69     ## Critical Rules
70
71         - NEVER make eligibility decisions yourself - ALWAYS
use evaluate_eligibility_tool
72         - NEVER call fill_form_tool for forms not in **Eligible
** list
73         - NEVER make claims that you have begun any form of
ongoing work, such as form filling
74         - NEVER claim to have completed a form, as the fill
form tool will provide this feedback when necessary
75         - NEVER try to determine eligibility yourself
76         - ALWAYS trust the eligibility results, or call
evaluate_eligibility_tool if more information has been
provided
77         - If a form is INELIGIBLE, clearly state it's
ineligible and explain why - NEVER suggest filling it
78         - FORM SELECTION PHASE: Present forms, ask which one,
NO tools
79         - Missing fields: Ask conversationally, don't call
fill_form_tool until user responds
80         - New situation info: Always call
evaluate_eligibility_tool first
81         - One tool at a time: Wait for results before next tool
call
```

```
82         - NEVER suggest or prompt for forms that are in the **  
Completed** list above  
83         - ONLY suggest forms from **Eligible** or **Uncertain**  
lists that are not completed  
84         - EXCEPTION: If user explicitly requests to redo/refill  
a completed form, then allow fill_form_tool for that form  
85         - If user asks about a completed form without  
explicitly requesting to redo it, remind them it's already  
done and offer alternatives  
86  
87         **Enhanced Guidance Rule:**  
88         - When eligibility results show forms that need minimal  
additional information, proactively guide the user toward  
providing that information rather than asking generic  
questions  
89         - Use the specific missing_info from eligibility  
results to ask targeted questions  
90         - When forms are ineligible for clear reasons (  
especially those relevant to user's needs), explain the  
ineligibility reason before suggesting alternatives  
91         - Don't ignore ineligible forms - acknowledge them and  
explain why they're not available - especially when the user  
is asking for other forms but none are forthcoming  
92     """
```

LISTING C.5: Chatbot System Prompt

Additional context is injected to provide information surrounding eligibility status and next steps.

```
1 def build_context(state: ChatState) -> str:  
2     parts = []
```

```
3
4     # Current phase indicators
5     if state.get("evaluation_complete") and state.get("
6     eligible_forms") and not state.get("current_form_name"):
7         forms = "\n".join(f"{i+1}. {f}" for i, f in enumerate(
8         state["eligible_forms"]))
9         parts.append(f"**FORM SELECTION PHASE**\nEligible forms
10        :\n{forms}\n\nPresent these forms and ask which one to fill.
11        DO NOT use tools.")
12
13    # Active form
14    if current := state.get("current_form_name"):
15        parts.append(f"**Currently filling:** {current}")
16
17    # Missing fields
18    if missing := state.get("missing_fields"):
19        fields = "\n".join(f"- {f}" for f in missing)
20        parts.append(f"**Missing fields (ask for ALL in one
21        message):**\n{fields}")
22
23    # Eligible forms
24    if eligible := state.get("eligible_forms"):
25        completed = state.get("completed_forms", [])
26        remaining = [f for f in eligible if f not in completed]
27        if remaining:
28            parts.append(f"**Eligible:** {' '.join(remaining)}
29            ")
30        if completed:
31            parts.append(f"**Completed:** {' '.join(completed)
32            }")
```

```
26
27     # Uncertain forms
28     if uncertain := state.get("uncertain_forms"):
29         uncertain_list = []
30         for form in uncertain:
31             name = form.get("form_name", "")
32             missing_info = form.get("missing_info", [])
33             if missing_info:
34                 uncertain_list.append(f"- {name}: needs {'', '}.
35 join(missing_info)}")
36             if uncertain_list:
37                 parts.append(f"**Uncertain (need more info):**\n" +
38 "\n".join(uncertain_list) +
39 "\n\nFocus on the form that BEST MATCHES the
40 user's stated needs. Ask if they'd like to explore that one,
41 or mention they can choose another.")
42
43     # Ineligible forms
44     if results := state.get("eligibility_results"):
45         ineligible = []
46         for name, assessment in results.items():
47             if isinstance(assessment, dict):
48                 if not assessment.get("eligible") and not
49 assessment.get("needs_more_info"):
50                     reason = assessment.get("reasoning", "")
51                     ineligible.append(f"- {name}: {reason}")
52         if ineligible:
53             parts.append(f"**Ineligible:**\n" + "\n".join(
54 ineligible))
```

50

51

```
return "\n\n".join(parts) if parts else ""
```

LISTING C.6: Construction of additional context based on internal state to be injected  
into chatbot prompt

# Appendix D

## JSON Coordinate Mapping for Form Fields

### D.1 Json Schema for Deterministic Field Mapping

The following snippet is the complete SCHEMA.json used within the system for deterministic field mapping.

```
1  {
2    "Local-Mind-Triage-Request.pdf": {
3      "form": "./forms/Local-Mind-Triage-Request.pdf",
4      "fields": [
5        {
6          "description": "Full Name",
7          "top_left_x": 137,
8          "top_left_y": 685,
9          "bottom_right_x": 327,
10         "bottom_right_y": 653,
11         "text": "",
12         "format": "words",
13         "page": 0
14       },
15       {
16         "description": "Date of birth",
17         "top_left_x": 152,
```

```
18         "top_left_y": 635,
19         "bottom_right_x": 336,
20         "bottom_right_y": 606,
21         "text": "",
22         "format": "DD/MM/YYYY",
23         "page": 0
24     },
25     {
26         "description": "Email Address",
27         "top_left_x": 161,
28         "top_left_y": 561,
29         "bottom_right_x": 337,
30         "bottom_right_y": 521,
31         "text": "",
32         "format": "email",
33         "page": 0
34     },
35     {
36         "description": "Address",
37         "top_left_x": 122,
38         "top_left_y": 482,
39         "bottom_right_x": 335,
40         "bottom_right_y": 437,
41         "text": "",
42         "format": "address, sans postcode",
43         "page": 0
44     },
45     {
46         "description": "Postcode",
47         "top_left_x": 131,
48         "top_left_y": 438,
49         "bottom_right_x": 294,
50         "bottom_right_y": 394,
51         "text": "",
52         "format": "Uk postcode format",
53         "page": 0
54     }
55 ]
56 },
57 "Primary-Health-Support-Referral.pdf": {
58     "form": "./forms/Primary-Health-Support-Referral.pdf",
59     "fields": [
60         {
```

```
61         "description": "NHS Number",
62         "top_left_x": 238,
63         "top_left_y": 651,
64         "bottom_right_x": 500,
65         "bottom_right_y": 625,
66         "text": "",
67         "format": "UK NHS number format",
68         "page": 0
69     },
70     {
71         "description": "Area of need",
72         "text": "",
73         "format": "Select from the following options",
74         "page": 0,
75         "choices": [
76             {
77                 "text": "counselling/ therapy",
78                 "top_left_x": 265,
79                 "top_left_y": 568,
80                 "bottom_right_x": 284,
81                 "bottom_right_y": 551
82             },
83             {
84                 "text": "peer support group",
85                 "top_left_x": 266,
86                 "top_left_y": 531,
87                 "bottom_right_x": 282,
88                 "bottom_right_y": 515
89             },
90             {
91                 "text": "financial support",
92                 "top_left_x": 261,
93                 "top_left_y": 487,
94                 "bottom_right_x": 286,
95                 "bottom_right_y": 469
96             },
97             {
98                 "text": "general support",
99                 "top_left_x": 266,
100                 "top_left_y": 444,
101                 "bottom_right_x": 282,
102                 "bottom_right_y": 427
103             }
104         ]
105     }
106 ]
```

```
104         ]
105     },
106     {
107         "description": "Has a current GP referral",
108         "top_left_x": 238,
109         "top_left_y": 345,
110         "bottom_right_x": 359,
111         "bottom_right_y": 323,
112         "text": "",
113         "format": "Yes or No",
114         "page": 0
115     },
116     {
117         "description": "Also applying for NHS low-income scheme?",
118         "top_left_x": 239,
119         "top_left_y": 234,
120         "bottom_right_x": 360,
121         "bottom_right_y": 211,
122         "text": "",
123         "format": "Yes or No",
124         "page": 0
125     },
126     {
127         "description": "Preferred language for service?",
128         "top_left_x": 240,
129         "top_left_y": 171,
130         "bottom_right_x": 509,
131         "bottom_right_y": 146,
132         "text": "",
133         "format": "word or words",
134         "page": 0
135     }
136 ]
137 },
138 "Health-Cost-Exemption.pdf": {
139     "form": "../forms/Health-Cost-Exemption.pdf",
140     "fields": [
141         {
142             "description": "First Name",
143             "top_left_x": 103,
144             "top_left_y": 663,
145             "bottom_right_x": 289,
146             "bottom_right_y": 639,
```

```
147         "text": "",
148         "format": "word",
149         "page": 0
150     },
151     {
152         "description": "Last Name",
153         "top_left_x": 306,
154         "top_left_y": 663,
155         "bottom_right_x": 492,
156         "bottom_right_y": 638,
157         "text": "",
158         "format": "word",
159         "page": 0
160     },
161     {
162         "description": "Contact Number",
163         "top_left_x": 101,
164         "top_left_y": 586,
165         "bottom_right_x": 287,
166         "bottom_right_y": 565,
167         "text": "",
168         "format": "UK Phone number format",
169         "page": 0
170     },
171     {
172         "description": "Email Address",
173         "top_left_x": 306,
174         "top_left_y": 586,
175         "bottom_right_x": 493,
176         "bottom_right_y": 563,
177         "text": "",
178         "format": "email",
179         "page": 0
180     },
181     {
182         "description": "Address Street",
183         "top_left_x": 102,
184         "top_left_y": 510,
185         "bottom_right_x": 494,
186         "bottom_right_y": 487,
187         "text": "",
188         "format": "First line of address, including house number",
189         "page": 0
```

```
190     },
191     {
192         "description": "Address Line 2",
193         "top_left_x": 102,
194         "top_left_y": 456,
195         "bottom_right_x": 493,
196         "bottom_right_y": 433,
197         "text": "",
198         "format": "Second line of address",
199         "page": 0
200     },
201     {
202         "description": "Town or City",
203         "top_left_x": 102,
204         "top_left_y": 402,
205         "bottom_right_x": 291,
206         "bottom_right_y": 379,
207         "text": "",
208         "format": "word or words",
209         "page": 0
210     },
211     {
212         "description": "Country",
213         "top_left_x": 306,
214         "top_left_y": 404,
215         "bottom_right_x": 494,
216         "bottom_right_y": 379,
217         "text": "",
218         "format": "only Wales or United Kingdom",
219         "page": 0
220     },
221     {
222         "description": "Postcode",
223         "top_left_x": 102,
224         "top_left_y": 347,
225         "bottom_right_x": 492,
226         "bottom_right_y": 324,
227         "text": "",
228         "format": "UK postcode format",
229         "page": 0
230     },
231     {
232         "description": "Annual Household Income",
```

```
233         "top_left_x": 94,
234         "top_left_y": 741,
235         "bottom_right_x": 286,
236         "bottom_right_y": 719,
237         "text": "",
238         "format": "Number",
239         "page": 1
240     },
241     {
242         "description": "Do you have savings or investments over £16000",
243         "text": "",
244         "format": "Select from the following options",
245         "page": 1,
246         "choices": [
247             {
248                 "text": "Yes",
249                 "top_left_x": 99,
250                 "top_left_y": 679,
251                 "bottom_right_x": 113,
252                 "bottom_right_y": 672
253             },
254             {
255                 "text": "No",
256                 "top_left_x": 99,
257                 "top_left_y": 659,
258                 "bottom_right_x": 110,
259                 "bottom_right_y": 652
260             }
261         ]
262     },
263     {
264         "description": "Occupation Status",
265         "text": "",
266         "format": "Select from the following options",
267         "page": 1,
268         "choices": [
269             {
270                 "text": "Employed",
271                 "top_left_x": 99,
272                 "top_left_y": 606,
273                 "bottom_right_x": 113,
274                 "bottom_right_y": 597
275             },
```

```
276         {
277             "text": "Unemployed",
278             "top_left_x": 99,
279             "top_left_y": 586,
280             "bottom_right_x": 110,
281             "bottom_right_y": 578
282         },
283         {
284             "text": "Retired",
285             "top_left_x": 99,
286             "top_left_y": 569,
287             "bottom_right_x": 110,
288             "bottom_right_y": 558
289         },
290         {
291             "text": "Student",
292             "top_left_x": 99,
293             "top_left_y": 548,
294             "bottom_right_x": 110,
295             "bottom_right_y": 542
296         },
297         {
298             "text": "Carer",
299             "top_left_x": 99,
300             "top_left_y": 531,
301             "bottom_right_x": 110,
302             "bottom_right_y": 523
303         }
304     ]
305 },
306 {
307     "description": "Chronic condition you experience",
308     "text": "",
309     "format": "Select from the following options",
310     "page": 2,
311     "choices": [
312         {
313             "text": "Diabetes",
314             "top_left_x": 99,
315             "top_left_y": 738,
316             "bottom_right_x": 113,
317             "bottom_right_y": 729
318         },
```

```
319         {
320             "text": "Epilepsy",
321             "top_left_x": 99,
322             "top_left_y": 718,
323             "bottom_right_x": 113,
324             "bottom_right_y": 708
325         },
326         {
327             "text": "Arthritis",
328             "top_left_x": 99,
329             "top_left_y": 701,
330             "bottom_right_x": 113,
331             "bottom_right_y": 692
332         },
333         {
334             "text": "Other",
335             "top_left_x": 99,
336             "top_left_y": 680,
337             "bottom_right_x": 112,
338             "bottom_right_y": 673
339         }
340     ]
341 },
342 {
343     "description": "Prescribed medications",
344     "top_left_x": 99,
345     "top_left_y": 630,
346     "bottom_right_x": 494,
347     "bottom_right_y": 374,
348     "text": "",
349     "format": "Comma separated list",
350     "page": 2
351 },
352 {
353     "description": "Declaration as an ordinary resident in Wales",
354     "text": "Yes",
355     "format": "Only Yes. Text is filled in, do not ask.",
356     "page": 2,
357     "choices": [
358         {
359             "text": "Yes",
360             "top_left_x": 101,
361             "top_left_y": 329,
```

```

362         "bottom_right_x": 112,
363         "bottom_right_y": 321
364     }
365 ]
366 }
367 ]
368 },
369 "Specialised-Mobility-Aid-Request.pdf": {
370     "form": "../forms/Specialised-Mobility-Aid-Request.pdf",
371     "fields": [
372         {
373             "description": "Full Name",
374             "top_left_x": 73,
375             "top_left_y": 658,
376             "bottom_right_x": 258,
377             "bottom_right_y": 631,
378             "text": "",
379             "format": "words",
380             "page": 0
381         },
382         {
383             "description": "Is a supporting physiotherapist or occupational therapist willing
↪ to sign off on this request?",
384             "text": "",
385             "format": "Select from the following options",
386             "page": 0,
387             "choices": [
388                 {
389                     "text": "yes",
390                     "top_left_x": 67,
391                     "top_left_y": 186,
392                     "bottom_right_x": 91,
393                     "bottom_right_y": 163
394                 },
395                 {
396                     "text": "no",
397                     "top_left_x": 152,
398                     "top_left_y": 180,
399                     "bottom_right_x": 175,
400                     "bottom_right_y": 167
401                 }
402             ]
403         },

```

```
404     {
405         "description": "Date of birth",
406         "top_left_x": 73,
407         "top_left_y": 607,
408         "bottom_right_x": 212,
409         "bottom_right_y": 583,
410         "text": "",
411         "format": "UK date format",
412         "page": 0
413     },
414     {
415         "description": "NHS Number",
416         "top_left_x": 73,
417         "top_left_y": 558,
418         "bottom_right_x": 277,
419         "bottom_right_y": 530,
420         "text": "",
421         "format": "UK NHS number format",
422         "page": 0
423     },
424     {
425         "description": "Current mobility aids in use",
426         "top_left_x": 72,
427         "top_left_y": 432,
428         "bottom_right_x": 444,
429         "bottom_right_y": 382,
430         "text": "",
431         "format": "Comma separated list",
432         "page": 0
433     },
434     {
435         "description": "Specialised mobility aids requested",
436         "top_left_x": 70,
437         "top_left_y": 356,
438         "bottom_right_x": 414,
439         "bottom_right_y": 300,
440         "text": "",
441         "format": "Comma separated list",
442         "page": 0
443     },
444     {
445         "description": "Daily mobility score",
446         "top_left_x": 70,
```

```

447         "top_left_y": 253,
448         "bottom_right_x": 202,
449         "bottom_right_y": 226,
450         "text": "",
451         "format": "Numerical score of 1 to 10. 1 Indicating immobility, 10 indicating high
↪ independence. If a mobility score is not directly mentioned, try to infer a score. If a
↪ score cannot be confidently inferred, then ask follow up questions.",
452         "page": 0
453     }
454 ]
455 },
456 "CHC-Initial-Screening.pdf": {
457     "form": "../forms/CHC-Initial-Screening.pdf",
458     "fields": [
459         {
460             "description": "Surname",
461             "top_left_x": 93,
462             "top_left_y": 445,
463             "bottom_right_x": 420,
464             "bottom_right_y": 420,
465             "text": "",
466             "format": "word",
467             "page": 0
468         },
469         {
470             "description": "All other names, sans surname",
471             "top_left_x": 93,
472             "top_left_y": 356,
473             "bottom_right_x": 420,
474             "bottom_right_y": 335,
475             "text": "",
476             "format": "word",
477             "page": 0
478         },
479         {
480             "description": "Residential status",
481             "text": "",
482             "format": "Select from the following options",
483             "page": 0,
484             "choices": [
485                 {
486                     "text": "own home",
487                     "top_left_x": 99,

```

```

488         "top_left_y": 235,
489         "bottom_right_x": 119,
490         "bottom_right_y": 226
491     },
492     {
493         "text": "residential care",
494         "top_left_x": 101,
495         "top_left_y": 211,
496         "bottom_right_x": 115,
497         "bottom_right_y": 205
498     },
499     {
500         "text": "hospital ward",
501         "top_left_x": 98,
502         "top_left_y": 190,
503         "bottom_right_x": 117,
504         "bottom_right_y": 180
505     }
506 ]
507 },
508 {
509     "description": "Mobility support need score",
510     "text": "",
511     "format": "Select a number from 1 to 6. 1 being totally independent and 6 being
↪ totally dependent.",
512     "page": 1,
513     "choices": [
514         {
515             "text": "1",
516             "top_left_x": 125,
517             "top_left_y": 465,
518             "bottom_right_x": 135,
519             "bottom_right_y": 460
520         },
521         {
522             "text": "2",
523             "top_left_x": 150,
524             "top_left_y": 465,
525             "bottom_right_x": 160,
526             "bottom_right_y": 460
527         },
528         {
529             "text": "3",

```

```
530         "top_left_x": 175,
531         "top_left_y": 465,
532         "bottom_right_x": 185,
533         "bottom_right_y": 460
534     },
535     {
536         "text": "4",
537         "top_left_x": 200,
538         "top_left_y": 465,
539         "bottom_right_x": 210,
540         "bottom_right_y": 460
541     },
542     {
543         "text": "5",
544         "top_left_x": 225,
545         "top_left_y": 465,
546         "bottom_right_x": 235,
547         "bottom_right_y": 460
548     },
549     {
550         "text": "6",
551         "top_left_x": 250,
552         "top_left_y": 465,
553         "bottom_right_x": 260,
554         "bottom_right_y": 460
555     }
556 ]
557 },
558 {
559     "description": "Communication support need score",
560     "text": "",
561     "format": "Select a number from 1 to 6. 1 being totally independent and 6 being
↪ totally dependent.",
562     "page": 1,
563     "choices": [
564         {
565             "text": "1",
566             "top_left_x": 125,
567             "top_left_y": 300,
568             "bottom_right_x": 135,
569             "bottom_right_y": 295
570         },
571         {
```

```
572         "text": "2",
573         "top_left_x": 150,
574         "top_left_y": 300,
575         "bottom_right_x": 160,
576         "bottom_right_y": 295
577     },
578     {
579         "text": "3",
580         "top_left_x": 175,
581         "top_left_y": 300,
582         "bottom_right_x": 185,
583         "bottom_right_y": 295
584     },
585     {
586         "text": "4",
587         "top_left_x": 200,
588         "top_left_y": 300,
589         "bottom_right_x": 210,
590         "bottom_right_y": 295
591     },
592     {
593         "text": "5",
594         "top_left_x": 225,
595         "top_left_y": 300,
596         "bottom_right_x": 235,
597         "bottom_right_y": 295
598     },
599     {
600         "text": "6",
601         "top_left_x": 250,
602         "top_left_y": 300,
603         "bottom_right_x": 260,
604         "bottom_right_y": 295
605     }
606 ]
607 },
608 {
609     "description": "Psychological support need score",
610     "text": "",
611     "format": "Select a number from 1 to 6. 1 being totally independent and 6 being
        ↪ totally dependent.",
612     "page": 1,
613     "choices": [
```

```
614         {
615             "text": "1",
616             "top_left_x": 125,
617             "top_left_y": 135,
618             "bottom_right_x": 135,
619             "bottom_right_y": 130
620         },
621         {
622             "text": "2",
623             "top_left_x": 150,
624             "top_left_y": 135,
625             "bottom_right_x": 160,
626             "bottom_right_y": 130
627         },
628         {
629             "text": "3",
630             "top_left_x": 175,
631             "top_left_y": 135,
632             "bottom_right_x": 185,
633             "bottom_right_y": 130
634         },
635         {
636             "text": "4",
637             "top_left_x": 200,
638             "top_left_y": 135,
639             "bottom_right_x": 210,
640             "bottom_right_y": 130
641         },
642         {
643             "text": "5",
644             "top_left_x": 225,
645             "top_left_y": 135,
646             "bottom_right_x": 235,
647             "bottom_right_y": 130
648         },
649         {
650             "text": "6",
651             "top_left_x": 250,
652             "top_left_y": 135,
653             "bottom_right_x": 260,
654             "bottom_right_y": 130
655         }
656     ]
```

```
657     },
658     {
659         "description": "Continence frequency of support required",
660         "top_left_x": 91,
661         "top_left_y": 541,
662         "bottom_right_x": 421,
663         "bottom_right_y": 513,
664         "text": "",
665         "format": "Free text to explain level of required support.",
666         "page": 2
667     },
668     {
669         "description": "I confirm that this reflects the last 7 days of care needs",
670         "text": "",
671         "format": "Only yes. If any other answer is inferred, then the applicant is not
        ↪ eligible for this form.",
672         "page": 2,
673         "choices": [
674             {
675                 "text": "yes",
676                 "top_left_x": 98,
677                 "top_left_y": 421,
678                 "bottom_right_x": 110,
679                 "bottom_right_y": 412
680             }
681         ]
682     }
683 ]
684 }
685 }
```

LISTING D.1: JSON Coordinate Field Mapping

## D.2 Python Code to Map Filled Form to Annotated PDF

```
1  def create_annotated_pdf(
2      filled_form: FormSchema,
3      form_name: str | None = None,
4      input_pdf_path: str | None = None,
```

```
5     output_pdf_path: str = "out.pdf"
6 ):
7     """
8     Creates an annotated PDF from the filled form data.
9     """
10    try:
11        input_pdf_path = filled_form.form
12
13        input_pdf_path = str(resolve_path(input_pdf_path))
14
15        if not os.path.exists(input_pdf_path):
16            return f"ERROR: Input PDF '{input_pdf_path}'
17            not found. Cannot generate output file. Please make sure '{
18            input_pdf_path}' is in the current directory."
19
20
21        reader = PdfReader(input_pdf_path)
22        writer = PdfWriter()
23
24        for page in reader.pages:
25            writer.add_page(page)
26
27        filled_fields = filled_form.fields
28
29        for field in filled_fields:
30            page_number = field.page
31            text_to_write = field.text.strip()
32            if not text_to_write:
33                continue
34
35            annotation = None
```

```
33         rect=None
34         text=""
35
36         if field.choices:
37             selected_choice = next(
38                 (c for c in field.choices if c.text.
39                  lower() == text_to_write.lower()),
40                 None
41             )
42
43             if selected_choice:
44                 rect = (
45                     selected_choice.top_left_x,
46                     selected_choice.bottom_right_y,
47                     selected_choice.bottom_right_x,
48                     selected_choice.top_left_y
49                 )
50
51                 # using dash to "tick" box or select
52                 choice, guess it's good enough for now
53                 text="-"
54
55             else:
56                 rect = (
57                     field.top_left_x,
58                     field.bottom_right_y,
59                     field.bottom_right_x,
60                     field.top_left_y
61                 )
62
63                 text=text_to_write
```

```
61
62         if rect is None:
63             print(f"Rect is None for field: {field.
64 description}, skipping...")
65             continue
66
67         print(f"Writing text: {text} to rect: {rect}")
68
69         annotation = FreeText(
70             text=text,
71             rect=rect,
72             font="Helvetica",
73             bold=True,
74             font_size="14pt",
75             font_color="000000",
76             background_color=None,
77             border_color=None
78         )
79
80         # printable
81         annotation.flags = 4
82
83         if annotation:
84             writer.add_annotation(page_number=
85 page_number, annotation=annotation)
86
87         with open(output_pdf_path, "wb") as output_stream:
88             writer.write(output_stream)
```

```
88         return f"Successfully created annotated PDF: {  
output_pdf_path}. All available information has been marked  
on the form."  
89  
90     except Exception as e:  
91         return f"ERROR: Failed to generate PDF. Check if '{  
input_pdf_path}' is a valid PDF and if coordinates are  
correct: {e}"
```

LISTING D.2: Form Filler System Message

## D.3 Annotated PDF Generation Ready for User Download

```
1  def generate_pdf_node(state: ChatState):  
2      """  
3      This node generates the PDF for the current completed form  
4      """  
5  
6      session_id = state.get("session_id")  
7      start_time = time.time()  
8  
9      current_form_name = state.get("current_form_name")  
10  
11     if not current_form_name:  
12         return create_error(  
13             state,  
14             "No form selected and no forms available."  
15         )
```

```
16
17     form_path = get_form_schema_by_name(current_form_name).form
18     if not form_path:
19         raise ValueError(f"Form path not found for {
current_form_name}")
20
21     filled_form = state.get("filled_form")
22     if not filled_form:
23         raise ValueError("Filled form not found")
24
25     form_base_name = os.path.splitext(current_form_name)[0]
26
27     forms_dir = get_session_forms_dir(session_id)
28     output_pdf_path = str(forms_dir / f"{form_base_name}_filled
.pdf")
29     pdf_result = create_annotated_pdf(filled_form, form_name=
form_path, output_pdf_path=output_pdf_path)
30
31     missing_fields = [f.description for f in filled_form.fields
if not f.text.strip()]
32
33     duration = time.time() - start_time
34     missing_fields_count = len(missing_fields)
35
36     output_paths = state.get("output_paths", [])
37     if output_pdf_path not in output_paths:
38         output_paths = output_paths + [output_pdf_path]
39
40     updated_state = mark_form_completed(state,
current_form_name)
```

```
41     updated_state = update_state(  
42         updated_state,  
43         current_form_name=None,  
44         info_request_count=0,  
45         filled_form=None,  
46         missing_fields=[],  
47         output_paths=output_paths  
48     )  
49  
50     if "ERROR" in pdf_result:  
51         log_error(session_id, f"Failed to generate PDF: {  
52             pdf_result}")  
53  
54         response_text = f"**I successfully filled the form data  
55         but failed to generate the PDF:** {pdf_result}. Please try  
56         again."  
57     else:  
58         log_form_generation(session_id, duration,  
59         current_form_name, output_pdf_path, missing_fields_count)  
60  
61         response_text = f"Done! Your {current_form_name} is  
62         ready for download below. "  
63  
64         response_text += "Remember to review and submit it  
65         through the appropriate channels."  
66  
67         if missing_fields:  
68             response_text += f" (Note: {'', ' '.join(  
69                 missing_fields)} left -blankfill manually if needed.)"
```

```
63     response_text += "\n\nAnything else I can help with?  
You can continue the conversation or share more about your  
situation and I will reevaluate your eligibility for other  
support!"  
64  
65     return update_state(  
66         updated_state,  
67         messages=[AIMessage(content=response_text)]  
68     )
```

LISTING D.3: Form Filler System Message

# Appendix E

## Pre-Post Intervention Survey Responses

This appendix contains the raw data collected during the pre-post intervention study ( $n = 8$ ). The data is presented as raw csv due to the volume of data, making formatting over tables difficult. The data uses ”|” as a delimiter as the raw strings contain commas. If there are any difficulties in parsing or dealing with this data, please let me know and I can provide the raw file.

### E.1 RAW CSV of Participant Survey Results

If I knew I were eligible, I would be willing to apply for support from the Welsh government or a charitable organisation|I believe that support services are relevant to people in my situation|I feel confident about successfully navigating the process of applying for support|Even if I were unsure of my eligibility, I would still invest time in seeking information about support available to me|I would recommend applying for support services to someone in a similar situation to mine|The idea of filling out complex application

forms makes me feel anxious|I often find the language in official support documentation to be confusing|I worry that I will submit incorrect information when applying for support|I believe the time required to complete support applications is a significant barrier|I am concerned about being judged for seeking support|I generally trust automated systems to handle my personal information securely|I prefer using an automated system over speaking with a human advisor for simple tasks|I believe an AI-powered system could accurately assess my eligibility for support|I would feel comfortable sharing sensitive information with an automated system|I would be willing to use an AI-powered system to help me apply for support|Have you ever applied for support services (government, charitable, or healthcare) in the past?|If yes, how would you describe your experience (Optional)|If you have NOT applied for support in the past despite potentially being eligible, what were the main barriers?|Please enter here if you selected "Other" above|Are you a current resident of Wales?|What is your age range?|What is the highest educational level you have achieved?|How would you rate your general comfort level with technology?|Have you ever used AI-powered chatbots or virtual assistants before? (e.g., ChatGPT, Alexa, Siri)|If I knew I were eligible, I would be willing to apply for support from the Welsh government or a charitable organisation|I believe that support services are relevant to people in my situation|I feel confident about successfully navigating the process of applying for support|Even if I were unsure of my eligibility, I would still invest time in seeking information about support available to me|I would recommend applying for support services to someone in a similar situation to mine|I found it easy to interact with the system|Learning to use the system was easy for me|It was easy to submit both messages and supporting documentation to the system|The system was clear in guiding me through the eligibility check, form selection, and form filling process|Using the system increased my understanding of my potential eligibility for support|The system was effective at handling the supporting documentation I provided|The system would help me to complete application forms more quickly than doing them manually|I found the system to be a useful tool for engaging with Welsh support services|I trust the eligibility decisions provided by the system|I felt confident

that the system securely handled the documents I uploaded|The system provides an accurate and unbiased assessment|I would be comfortable submitting an application form generated by the system|If the system were available today, how likely would you be to use it when seeking support in the future?|I would recommend the system to others who need to apply for support|What did you find to be the most helpful feature or aspect of the system?|What was the most frustrating part of using the system, and how could it be improved?|Please share any additional thoughts about how this system might impact access to support services in Wales 5|5|4|3|5|5|6|7|7|5|2|6|2|2|5|Yes|long and confusing. Would usually call the service to ask for guidance/ elaborate on some of the information the system is asking for.|||Yes|25-34|Bachelor's degree|Somewhat comfortable|Frequently|5|5|6|6|5|5|6|5|7|7|5|7|7|4|4|5|5|5|6|the finding and filling out of the documents needed. as well as asking for specific documents to provide it.|understanding if my documents were uploaded. Maybe when submitting the documents the bot should say "evidence taken" rather than it being with clicking submit prompt |I think it provides quick and easy access to information that might be difficult to find in recent times. especially due to the difficulty of working with modern search engines to find where to go and get information needed. the quick and easy access makes it easier to check for eligibility even if Id think I might not need it 7|4|3|3|4|5|6|6|7|7|4|6|6|6|7|Yes|Judgemental and slow|Not applicable||Yes|25-34|Bachelor's degree|Very comfortable|Rarely|7|7|7|7|7|7|7|7|7|7|7|7|7|7|7|I was unable to find specific answers to questions in the supporting documents but the tool found them for me|There was nothing frustrating in using the system|This would make it significantly easier for people with difficulties in reading/writing/processing information to access support services without needing to understand the legal jargon that forms for support can sometimes contain. 6|6|4|2|4|2|1|2|2|1|2|1|6|2|6|Yes||Not applicable||Yes|25-34|Master's degree|Somewhat comfortable|Rarely|6|6|4|3|2|6|6|6|6|6|6|6|4|2|4|5|5|5|The speed/responsiveness is nice|I didn't find it frustrating|People with social anxiety might be nervous speaking to humans about their issues. AI depersonalises the experience and may be more comfortable on that account for certain people. 5|3|6|3|3|3|3|3|6|3|3|7|5|3|3|Yes|Long winded|Didn't know I was eligible;Process was too complicated;Didn't have time;Didn't

know where to start;Other (Please specify below)|Didn't feel worth the hassle to go through the process when I didn't know what I might gain from it|Yes|35-44|Bachelor's degree|Very comfortable|Occasionally|5|3|5|3|3|5|5|4|3|4|3|4|4|5|4|5|5|4|4|System knows what criteria to consider for eligibility - this will make it a lot easier to focus on providing the right information rather than spending time working that part out.|System did not indicate that it could accept document uploads and so, in persona, I would have missed out on this functionality|Being able to access this in your own time will be very valuable, and will take strain off the NHS and other in person services. Being able to interact through a guided process is going to allow people to focus on the information relevant to them rather than becoming lost and overwhelmed! 7|7|1|5|7|7|7|7|4|3|4|1|1|1|1|Yes|Very confusing and difficult as someone who suffers with Autism and ADHD. It took a years for me to acquire help through a support worker due to waiting lists.|Not applicable||Yes|25-34|A-Level, BTEC, or equivalent|Somewhat comfortable|Occasionally|7|7|4|7|7|5|5|4|7|5|5|7|7|5|4|7|6|7|7| is much quicker and clearer than going through traditional application processes when looking for help.|I got stuck in a loop when trying to find out if there was faster help I could access. It didn't remember that I already had a GP referral |Applying for assistance would be quicker but waiting lists themselves for help are still the main issue. This could also cause a backlog of applications possibly extending that wait list. It's useful for finding out what help you are eligible for and applying for it but that is not the main issue in my opinion. 7|4|6|2|2|5|7|5|2|6|5|3|2|6|5|No||Not applicable||Yes|25-34|Master's degree|Very comfortable|Very frequently|7|3|6|4|4|5|7|7|7|7|7|7|6|5|7|5|7|7|Help with eligibility checks and form filling. |Not very flexible in giving response if ineligible. |It is very helpful with making applications. I think it would encourage more people to apply. People who are less confident with technology etc. 5|2|5|4|6|5|3|6|5|7|1|2|2|1|2|Yes|Job-seekers Allowance|Not applicable||Yes|35-44|Master's degree|Very comfortable|Rarely|5|2|5|5|6|7|7|6|5|5| was able to pick out relevant information, even with typos|I don't think there was a frustrating part of the system|This could be useful for people unsure about how to go about finding support, even as a starting point even if they don't go on to use they system fully 5|6|3|5|5|7|6|7|7|7|6|7|2|6|4|No||Didn't know I was eligible;Didn't know where to

start||Yes|18-24|Master's degree|Very comfortable|Occasionally|7|5|6|7|6|7|7|6|7|6||7|7|4||5|4|6|6|Not  
needing to search through government websites myself and instead being able to get an  
idea of eligibility much faster.|It ticked one incorrect answer in the form without asking  
for clarification. This can be fixed by the user, but would be much easier if the chat-  
bot asked to make sure the information was correct.|I think it would increase knowledge  
of eligibility and make information easier to find, helping people who are in need of  
support

# Bibliography

- [1] Haolin Wu, Tim French, Wei Liu, and Melinda Hodkiewicz. Automatic semantic knowledge extraction from electronic forms. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 238(5):933–944, May 2022. ISSN 1748-0078. doi: 10.1177/1748006x221098272.
- [2] Mustafa Al Jnainati, Jana Al Jnainati, Shree Rath, Savvy Aujla, Eeshah Nasir, Aishwarya Govindarajan, Yatee Samantaray, and Mehak Semy. Transforming paperwork with ai: applications across healthcare and other industries. *AI amp; SOCIETY*, May 2025. ISSN 1435-5655. doi: 10.1007/s00146-025-02380-0.
- [3] Mehmet Yanit, Metehan Yanit, and Fang Wan. Right agent, wrong level of hedonism: How high (vs low) hedonic values in ai-performed tasks lead to decreased perceptions of humanlikeness, warmth, and less consumer support. *Computers in Human Behavior*, 147:107870, October 2023. ISSN 0747-5632. doi: 10.1016/j.chb.2023.107870.
- [4] Laurie Hughes, Yogesh K. Dwivedi, Tegwen Malik, Mazen Shawosh, Mousa Ahmed Albashrawi, Il Jeon, Vincent Dutot, Mandanna Appanderanda, Tom Crick, Rahul De’, Mark Fenwick, Senali Madugoda Gunaratnege, Paulius Jurcys, Arpan Kumar Kar, Nir Kshetri, Keyao Li, Sashah Mutasa, Spyridon Samothrakis, Michael Wade, and Paul Walton. Ai agents and agentic systems: A multi-expert analysis. *Journal of Computer Information Systems*, pages 1–29, April 2025. ISSN 2380-2057. doi: 10.1080/08874417.2025.2483832.

- [5] Siliang Chen, Xinbin Liang, Ying Liu, Xilin Li, Xinqiao Jin, and Zhimin Du. Customized large-scale model for human-ai collaborative operation and maintenance management of building energy systems. *Applied Energy*, 393:126169, September 2025. ISSN 0306-2619. doi: 10.1016/j.apenergy.2025.126169.
- [6] Jiangbo Yu. Preparing for an agentic era of human-machine transportation systems: Opportunities, challenges, and policy recommendations. *Transport Policy*, 171: 78–97, September 2025. ISSN 0967-070X. doi: 10.1016/j.tranpol.2025.05.030.
- [7] Yufan Chen, Ching Ting Leung, Bowen Yu, Jianwei Sun, Yong Huang, Linyan Li, Hao Chen, and Hanyu Gao. A multi-agent system enables versatile information extraction from the chemical literature, 2025.
- [8] Federico Lorenzo Barra, Giovanna Rodella, Alessandro Costa, Antonio Scalogna, Luca Carenzo, Alice Monzani, and Francesco Della Corte. From prompt to platform: an agentic ai workflow for healthcare simulation scenario design. *Advances in Simulation*, 10(1), May 2025. ISSN 2059-0628. doi: 10.1186/s41077-025-00357-z.
- [9] Seungmin Choi and Yuchul Jung. Knowledge graph construction: Extraction, learning, and evaluation. *Applied Sciences*, 15(7):3727, March 2025. ISSN 2076-3417. doi: 10.3390/app15073727.
- [10] Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, 13:18912–18936, 2025. ISSN 2169-3536. doi: 10.1109/access.2025.3532853.
- [11] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *IEEE Access*, 6:28573–28593, 2018. ISSN 2169-3536. doi: 10.1109/access.2018.2831228.

- [12] Alejandro Torreño, Eva Onaindia, Antonín Komenda, and Michal Štolba. Cooperative multi-agent planning: A survey. *ACM Computing Surveys*, 50(6):1–32, November 2017. ISSN 1557-7341. doi: 10.1145/3128584.
- [13] Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6), June 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae353.
- [14] Matthew Toles, Rattandeep Singh, and Isaac Song Zhou Yu. Formgym: Doing paperwork with agents, 2025.
- [15] Enayat Rajabi, Allu Niya George, and Karishma Kumar. The role of knowledge graphs in chatbots. *The Electronic Library*, 42(3):483–497, June 2024. ISSN 0264-0473. doi: 10.1108/el-03-2023-0066.
- [16] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D’amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, July 2021. ISSN 1557-7341. doi: 10.1145/3447772.
- [17] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568, August 2022. ISSN 2326-3865. doi: 10.1109/tkde.2020.3028705.
- [18] Haoze Yu, Haisheng Li, Dianhui Mao, and Qiang Cai. A relationship extraction method for domain knowledge graph construction. *World Wide Web*, 23(2): 735–753, January 2020. ISSN 1573-1413. doi: 10.1007/s11280-019-00765-y.

- [19] Minghui Wu and Xindong Wu. On big wisdom. *Knowledge and Information Systems*, 58(1):1–8, December 2018. ISSN 0219-3116. doi: 10.1007/s10115-018-1282-y.
- [20] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62, November 2023. ISSN 1557-7341. doi: 10.1145/3618295.
- [21] Feng Chen, Liqin Wang, Julie Hong, Jiaqi Jiang, and Li Zhou. Unmasking bias in artificial intelligence: a systematic review of bias detection and mitigation strategies in electronic health record-based models. *Journal of the American Medical Informatics Association*, 31(5):1172–1183, March 2024. ISSN 1527-974X. doi: 10.1093/jamia/ocae060.
- [22] Sarfraz Brohi, Qurat-ul-ain Mastoi, N. Z. Jhanjhi, and Thulasyammal Ramiah Pillai. A research landscape of agentic ai and large language models: Applications, challenges and future directions. *Algorithms*, 18(8):499, August 2025. ISSN 1999-4893. doi: 10.3390/a18080499.
- [23] Simon Knollmeyer, Oğuz Caymazer, and Daniel Grossmann. Document graphrag: Knowledge graph enhanced retrieval augmented generation for document question answering within the manufacturing domain. *Electronics*, 14(11):2102, May 2025. ISSN 2079-9292. doi: 10.3390/electronics14112102.
- [24] Bin Shi, Hao Wang, Yueyan Li, and Sanhong Deng. Relagraph: Improving embedding on small-scale sparse knowledge graphs by neighborhood relations. *Information Processing and Management*, 60(5):103447, September 2023. ISSN 0306-4573. doi: 10.1016/j.ipm.2023.103447.
- [25] Xiaohan Zou. A survey on application of knowledge graph. *Journal of Physics: Conference Series*, 1487(1):012016, March 2020. ISSN 1742-6596. doi: 10.1088/1742-6596/1487/1/012016.

- [26] Joëd Ngangmeni and Danda B. Rawat. Swamped with too many articles? graphrag makes getting started easy. *AI*, 6(3):47, March 2025. ISSN 2673-2688. doi: 10.3390/ai6030047.
- [27] Xu Ji, Luo Xu, Landi Gu, Junjie Ma, Zichao Zhang, and Wei Jiang. Raprag: A retrieval-augmented generation framework with adaptive retrieval task planning. *Electronics*, 14(21):4269, October 2025. ISSN 2079-9292. doi: 10.3390/electronics14214269.
- [28] Yao-Yang Liu, Zhen Zheng, Feng Zhang, Jin-Cheng Feng, Yi-Yang Fu, Ji-Dong Zhai, Bing-Sheng He, Xiao Zhang, and Xiao-Yong Du. A comprehensive taxonomy of prompt engineering techniques for large language models. *Frontiers of Computer Science*, 20(3), October 2025. ISSN 2095-2236. doi: 10.1007/s11704-025-50058-z.
- [29] Ana-Paula Correia, Sean Hickey, and Fan Xu. Realizing the possibilities of the large language models: Strategies for prompt engineering in educational inquiries. *Theory Into Practice*, 64(4):434–447, August 2025. ISSN 1543-0421. doi: 10.1080/00405841.2025.2528545.
- [30] Guoqing Wang, Zeyu Sun, Sixiang Ye, Zhihao Gong, Yizhou Chen, Yifan Zhao, Qingyuan Liang, and Dan Hao. Do advanced language models eliminate the need for prompt engineering in software engineering? *ACM Transactions on Software Engineering and Methodology*, October 2025. ISSN 1557-7392. doi: 10.1145/3771933.
- [31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

- URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf).
- [32] Philip Fei Wu, Charlotte Summers, Arjun Panesar, Amit Kaura, and Li Zhang. Ai hesitancy and acceptability—perceptions of ai chatbots for chronic health management and long covid support: Survey study. *JMIR Human Factors*, 11: e51086–e51086, July 2024. ISSN 2292-9495. doi: 10.2196/51086.
- [33] Md Imran Hossain, Ghada Zamzmi, Peter R. Mouton, Md Sirajus Salekin, Yu Sun, and Dmitry Goldgof. Explainable ai for medical data: Current methods, limitations, and future directions. *ACM Computing Surveys*, 57(6):1–46, February 2025. ISSN 1557-7341. doi: 10.1145/3637487.
- [34] Pooja Rao S B, Manish Agnihotri, and Dinesh Babu Jayagopi. Improving asynchronous interview interaction with follow-up question generation. *International Journal of Interactive Multimedia and Artificial Intelligence*, 6(5):79, 2021. ISSN 1989-1660. doi: 10.9781/ijimai.2021.02.010.
- [35] Matthias Hansen, André Pomp, Kemal Erki, and Tobias Meisen. Data-driven recognition and extraction of pdf document elements. *Technologies*, 7(3):65, September 2019. ISSN 2227-7080. doi: 10.3390/technologies7030065.
- [36] Cartic Ramakrishnan, Abhishek Patnia, Eduard Hovy, and Gully APC Burns. Layout-aware text extraction from full-text pdf of scientific articles. *Source Code for Biology and Medicine*, 7(1), May 2012. ISSN 1751-0473. doi: 10.1186/1751-0473-7-7.
- [37] PyPDF. Pypdf documentation. URL <https://pypdf.readthedocs.io/en/3.15.3/user/adding-pdf-annotations.html>.
- [38] Martin Hähnel. Ethical challenges and solutions in ai-driven medical data management: a focus on distributed machine learning. *Discover Artificial Intelligence*, 5(1), May 2025. ISSN 2731-0809. doi: 10.1007/s44163-025-00266-0.

- [39] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. *Explainable AI Methods - A Brief Overview*, pages 13–38. Springer International Publishing, 2022. ISBN 9783031040832. doi: 10.1007/978-3-031-04083-2\_2.
- [40] Akshita Rohatgi and Tae Jung Park. Privacy in the public: Analysing the eu framework to outline approaches for regulating ai personal data scraping. *Computer Law amp; Security Review*, 57:106150, July 2025. ISSN 2212-473X. doi: 10.1016/j.clsr.2025.106150.
- [41] Revathy Pulugu Sonam Mittal Dileep Pulugu S. Lokesh, Sudeshna Chakraborty and R. Muruganantham. Ai based big data analytics model for medical applications ai-based big data analytics model for medical applications. *Measurement. Sensors*, 24:100534, 2022. ISSN 2665-9174. URL [https://librarysearch.uwtsd.ac.uk/permalink/44WHELF\\_UWTSD/igoone/cdi\\_doaj\\_primary\\_oai\\_doaj\\_org\\_article\\_b45a12b8e271415c896120dabc759151](https://librarysearch.uwtsd.ac.uk/permalink/44WHELF_UWTSD/igoone/cdi_doaj_primary_oai_doaj_org_article_b45a12b8e271415c896120dabc759151).
- [42] Serge Autexier, Christoph Lüth, and Rolf Drechsler. *Das Bremen Ambient Assisted Living Lab und darüber hinaus – Intelligente Umgebungen, smarte Services und Künstliche Intelligenz in der Medizin für den Menschen*, pages 835–850. Springer Fachmedien Wiesbaden, 2022. ISBN 9783658335977. doi: 10.1007/978-3-658-33597-7\_40.
- [43] Karl Stoeger, David Schneeberger, Peter Kieseberg, and Andreas Holzinger. Legal aspects of data cleansing in medical ai. *Computer Law amp; Security Review*, 42: 105587, September 2021. ISSN 2212-473X. doi: 10.1016/j.clsr.2021.105587.
- [44] Abdullah Alanazi. Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, 30:100924, 2022. ISSN 2352-9148. doi: 10.1016/j.imu.2022.100924.
- [45] Richard E. Dunning, Baruch Fischhoff, and Alex L. Davis. When do humans heed ai agents’ advice? when should they? *Human Factors: The Journal of the Human*

- Factors and Ergonomics Society*, 66(7):1914–1927, August 2023. ISSN 1547-8181. doi: 10.1177/00187208231190459.
- [46] Rongbin Yang and Santoso Wibowo. User trust in artificial intelligence: A comprehensive conceptual framework. *Electronic Markets*, 32(4):2053–2077, November 2022. ISSN 1422-8890. doi: 10.1007/s12525-022-00592-6.
- [47] Enes Erul, Yusuf Aktekin, Furkan Berk Danisman, Sukru Armanc Gumustas, Busra Sarac Aktekin, Emre Yekeduz, and Yuksel Urun. Perceptions, attitudes, and concerns on artificial intelligence applications in patients with cancer. *Cancer Control*, 32, May 2025. ISSN 1526-2359. doi: 10.1177/10732748251343245.
- [48] Ilaria Querci, Camilla Barbarossa, Simona Romani, and Francesco Ricotta. Explaining how algorithms work reduces consumers’ concerns regarding the collection of personal data and promotes ai technology adoption. *Psychology amp; Marketing*, 39(10):1888–1901, June 2022. ISSN 1520-6793. doi: 10.1002/mar.21705.
- [49] Ivana Paccoud, Mayca Marín Valero, Laura Carrasco Marín, Noémi Bontridder, Alzhraa Ibrahim, Jüergen Winkler, Messaline Fomo, Stefano Sapienza, Fouad Khoury, Jean-Christophe Corvol, Holger Fröhlich, and Jochen Klucken. Patient perspectives on the use of digital medical devices and health data for ai-driven personalised medicine in parkinson’s disease. *Frontiers in Neurology*, 15, December 2024. ISSN 1664-2295. doi: 10.3389/fneur.2024.1453243.
- [50] Jie Wu. Social and ethical impact of emotional ai advancement: the rise of pseudo-intimacy relationships and challenges in human interactions. *Frontiers in Psychology*, 15, November 2024. ISSN 1664-1078. doi: 10.3389/fpsyg.2024.1410462.
- [51] Rebecca Kehat, Ron S. Hirschprung, and Shani Alkoby. Enhancing user acceptance of an ai agent’s recommendation in information-sharing environments. *Applied Sciences*, 14(17):7874, September 2024. ISSN 2076-3417. doi: 10.3390/app14177874.

- [52] Tae Hyun Baek, Hyoje Jay Kim, and Jungkeun Kim. Ai-generated recommendations: Roles of language style, perceived ai human-likeness, and recommendation agent. *International Journal of Hospitality Management*, 126:104106, April 2025. ISSN 0278-4319. doi: 10.1016/j.ijhm.2025.104106.
- [53] S Shyam Sundar. Rise of machine agency: A framework for studying the psychology of human–ai interaction (haii). *Journal of Computer-Mediated Communication*, 25(1):74–88, January 2020. ISSN 1083-6101. doi: 10.1093/jcmc/zmz026.
- [54] LangChain. Langgraph, . URL <https://www.langchain.com/langgraph>.
- [55] Amazon Web Services. Amazon q - generative ai assistant. URL <https://aws.amazon.com/q/>.
- [56] ChromaDB. Chromadb. URL <https://pypi.org/project/chromadb/>.
- [57] LangChain. Openaiembeddings, . URL [https://docs.langchain.com/oss/python/integrations/text\\_embedding/openai](https://docs.langchain.com/oss/python/integrations/text_embedding/openai).
- [58] LangChain. Text splitters, . URL <https://docs.langchain.com/oss/python/integrations/splitters>.
- [59] jol333.github.io. Pdf coordinates. URL <https://jol333.github.io/pdf-coordinates/>.
- [60] Gradio. Gradio app. URL <https://www.gradio.app/>.
- [61] Welsh Government. Written statement: Strategic artificial intelligence advisory group and welsh government office for ai, June 2025. URL <https://www.gov.wales/written-statement-strategic-artificial-intelligence-advisory-group-and-welsh>
- [62] HuggingFace. Hugging face spaces. URL <https://huggingface.co/spaces>.

- [63] Andrina Granić and Nikola Marangunić. Technology acceptance model in educational context: A systematic literature review. *British Journal of Educational Technology*, 50(5):2572–2593, July 2019. ISSN 1467-8535. doi: 10.1111/bjet.12864.
- [64] Viswanath Venkatesh and Hillol Bala. Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2):273–315, May 2008. ISSN 1540-5915. doi: 10.1111/j.1540-5915.2008.00192.x.
- [65] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*, CHI '90, pages 249–256. ACM Press, 1990. doi: 10.1145/97243.97281.
- [66] Jim Frost. Cohens d: Definition, using and examples. URL <https://statisticsbyjim.com/basics/cohens-d/>.