# IBM AI AR techdoc

Ani Bitri

October 13, 2025

## Contents

# 1 Problem Statement

## 1.1 Introduction and Context

Technical documentation plays a critical role in helping developers understand, implement, operate and maintain software systems. In enterprise environments, documentation is the primary interface between developers and complex systems, making clarity and accessibility essential. However, despite its importance, technical documentation remains predominantly static, text-based and difficult to navigate. As systems grow in complexity, dynamic processes such as data flows, dependencies and interactions are often poorly represented in traditional documentation formats. Consequently, many users often struggle to find the information they need, leading to frustration, errors and inefficiencies. This raises the need for innovative solutions that can enhance the user experience and improve comprehension of technical documentation.

## 1.2 Existing Technologies

There are several existing technologies which address parts of the problem but do not yet offer a comprehensive solution. These include:

- **Static and Web-Based Documentation:** Different Platforms like GitBook, ReadTheDocs and Markdown-based tools offer well-structured and accessible documentation but lack interactivity and visualization capabilities. Information is conveyed effectively through text and images but remains static and fails to envision the dynamic nature of modern software systems.
- **AI Assistants:** In the last few years, the rise of AI-powered assistants like ChatGPT and Gemini have been transformative. These tools can understand and interpret technical text, summarize content and answer user queries. Yet, these assistants operate in isolation from the documentation itself, requiring users to switch contexts and manually input queries, disconnecting them from visual elements and offering limited support for the understanding of system architecture and interactions.
- **Augmented Reality Applications:** Industrial AR tools, such as PTC Vuforia, Microsoft Dynamics 365 Guides and Scope AR, have been used to overlay digital information onto real-world objects. These solutions enhance spatial understanding and are particularly useful in maintenance and training contexts. However, they typically rely on specific and expensive hardware, such as the HoloLens, or complex 3D models, thus limiting their accessibility for general users.

## 1.3 Gaps in Current Solutions

Despite the advancements in AR and AI technologies, the current solutions exhibit several limitations:

- **Lack of Integration between AR and AI:** Visualization (AR) and comprehension (AI) are treated as separate domains. There is limited integration or implementation that combines the strengths of both to deliver a seamless user experience.
- **Limited Visual Comprehension:** Text and 2D diagrams are insufficient for conveying complex interactions and dynamic processes. Current documentation formats do not effectively utilize visual aids to enhance understanding.
- **Hardware and Platform Barriers:** Most AR implementations are dependent on specialized hardware, such as the HoloLens or AR glasses, which are costly or not widely available. This restricts the accessibility of AR-enhanced documentation to a broader audience.

## 1.4 Problem Definition

Given the limitations of existing documentation systems and the potential of emerging technologies, there is a need for an innovative solution which can be easily accessible, interactive and capable of enhancing the understanding of technical documentation. In order to address this need, the project aims to develop an AI-powered mobile application that uses AR technology to augment technical documentation. The proposed system must allow users to point their mobile devices at a technical diagram and view interactive AR overlays that explain components, relationships and interactions. At the same time, an integrated AI assistant, IBM Watson/Granite, will be available to answer user queries and provide additional context.

Therefore, this project aims to bridge the gap between visual comprehension and intelligent interaction in technical documentation, ultimately demonstrating how static documents can become interactive, explainable and user-friendly.

# 2 Project Overview

## 2.1 Purpose

Many developers face challenges while reading and understanding technical documentation. This project aims to create an AI-powered mobile application which, supported by AR technology, will enhance the user experience by providing interactive and immersive documentation. Powered by IBM Watson/Granite, the app will offer several features to assist developers in navigating and comprehending complex technical documents, including text recognition, interactive AR overlays, chatbot assistance and more.

## 2.2 Objectives within the scope

The primary objectives of this project are to:

1. Implement AR diagram augmentation
    - Detect and track diagrams in printed and digital forms.
    - Overlay interactive elements on diagrams to provide additional context and explanations.
2. Integrate an AI assistant
    - Employ IBM Watson/Granite to interpret the scanned documentation text and answer user queries.
    - Support natural language questions such as "What is the purpose of this diagram?" or "Explain this concept in simpler terms.".
3. Preserve accessibility and compliance
    - Keep the core document unchanged and externalize enhancements through AR overlays to comply with accessibility and legal requirements.
4. Develop a functional mobile prototype
    - Deliver a working mobile application prototype that demonstrates the key features and functionalities.
    - Conduct user testing to gather feedback and refine the application.
    - Provide a short demonstration.

## 2.3 Boundaries and Out-of-Scope Elements

To keep the project achievable within the given timeframe, the following elements are considered out of scope:

- Full production deployment or enterprise-level integration.
- Hardware-specific AR is excluded; the focus is on mobile devices.
- Cross-platform optimization beyond the primary target platform (e.g., iOS or Android).

## 2.4 Expected Deliverables

- A prototype mobile application demonstrating real-time recognition and overlay of technical documentation.
- Integrated AI assistant interface capable of answering user queries based on the documentation content.
- A comprehensive project report detailing the design, implementation, testing processes and evaluation results.
- VIVA presentation and demonstration.

## 2.5 Target Outcomes

- Enhanced user experience for developers interacting with technical documentation.
- Improved comprehension of complex technical concepts through interactive AR elements and AI assistance.
- A foundation for future development and potential commercialization of the application.

# 3 Requirements

Each requirement will be described in the format RnC/D, where n is the requirement number and C or D indicates whether the requirement is customer (C) or developer (D) oriented. For example, R1C refers to the first customer requirement, while R2D refers to the second developer requirement. Each requirement will be detailed with its description, priority, verification method and traceability.

## 3.1 Functional Requirements

List the key functionalities the software system must support. Provide clear and concise descriptions of features and interactions.

1. **Augmented Reality System**

## 3.2 Non-Functional Requirements

Outline performance, usability, reliability, and other quality attributes expected from the system.

# 4 System Architecture

Provide a high-level overview of the system architecture. Include diagrams where appropriate to illustrate the system components and their interactions.

## 4.1 Design Pattern

The design pattern which will be adopted for this project is the Model-View-ViewModel (MVVM) pattern. This pattern is well-suited for applications that require a clear separation of concerns between the user interface and logic, making it easier to manage and test the application. Furthermore, the MVVM pattern facilitates a more modular and maintainable codebase, which is essential for the iterative development process planned for this project. The image below illustrates the MVVM architecture which will be used in the project.

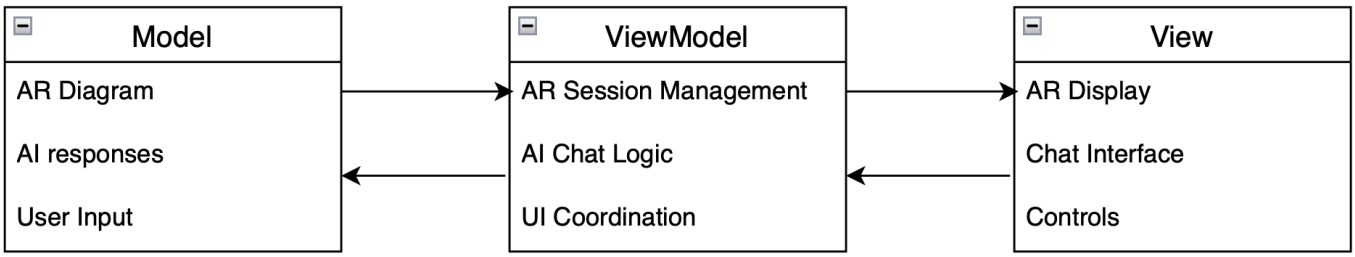


Figure 1: MVVM Architecture Pattern

## 4.2 Frontend Design

### 4.2.1 User Interface and User Experience

The application will feature a user-friendly interface that allows users to easily navigate through the app and access its features. The UI will be designed to be intuitive and responsive, ensuring a seamless user experience. When the user open the app, they will be presented with a simple home screen with options to scan a document, upload a document or access the history of previously scanned documents. Besides the home screen, the app will also include a tab for the chatbot, where users can interact with the AI assistant, and a settings tab for configuring app preferences. The image below illustrates the proposed UI design for the application:

Upon selecting the scan option, the user will be directed to the camera interface, where they can capture pictures of the technical documentation. Alternatively, the user can choose to upload the document from their device's storage and access the same features. After scanning or uploading a document, the app will process the text, images and diagrams, and give the user the option to view the AR overlays or interact with the AI assistant. The AR overlays will provide interactive elements that explain components, relationships and interactions within the diagrams based on the user's input and the information extracted from the document. The image below illustrates a flow chart of the app's user interface and navigation:

### 4.2.2 Error Handling and Feedback

To ensure a smooth user experience, the application will include robust error handling mechanisms. Internal errors will be handled gracefully and automatically, with appropriate messages displayed to the user upon interruptions. For example, if the app fails to recognize a diagram or if the AI assistant cannot process a query, the user will be informed of the issue and provided with suggestions for resolution. External errors will be handled by providing informative error messages to guide users in case of issues such as failed scans or uploads, network connectivity problems or AR tracking errors. Furthermore, error prevention strategies will be implemented to minimize their occurrence and impact on the user experience. Such strategies include input validation, network status checks and AR tracking optimizations. This approach will ensure that users can effectively utilize the app's features without being hindered by technical issues.

#### 4.2.3 Integration with Backend and Data Flow

The frontend of the application will communicate with the backend services to process the scanned or uploaded documents and retrieve relevant information. The data flows as follows:

1. The user scans or uploads a document using the app's interface.
2. The frontend validates the input and sends the document data to the backend for processing.
3. The validated data is sent to the backend services for text recognition, diagram analysis and AI processing.
4. The backend processes the data and generates AR overlays and AI responses based on the document content.
5. The generated AR overlays and AI responses are sent back to the frontend for display to the user.
6. The user interacts with the AR overlays and AI assistant, and any further queries or actions are sent back to the backend for processing.

### 4.3 Backend Design

#### 4.3.1

#### 4.3.2

#### 4.3.3

### 4.4 Technology Stack

The project will utilize the following technologies:

- **Frontend:** React Native for cross-platform mobile development, ARCore/ARKit/ViroReact for augmented reality functionalities.
- **Backend:** Node.js with Express for server-side logic, IBM Granite 4.0 for AI capabilities, IBM Granite Vision for image recognition, and MongoDB for data storage.
- **Development Tools:** Visual Studio Code for code editing, Git for version control, REST APIs for communication between frontend and backend.

## 5 Development Plan and Project Philosophy

### 5.1 Methodology

Given the dynamic nature of the project and the need for iterative development, the Scrum framework within the Agile methodologies will be adopted. The approach will allow for flexibility, continuous feedback and incremental delivery of features. Weekly stand-up meetings will be held to discuss progress and address any challenges. The work will be organized into sprints, each lasting approximately one to two weeks, with clearly defined goals. At the end of each sprint, a review session will be held with the assessors to demonstrate progress and gather feedback.

## 5.2 Time Management and Milestones

The Gantt chart below outlines the planned work schedule, including key milestones and deadlines. Weekly meetings with the assessors will also be scheduled to ensure alignment and address any issues promptly.

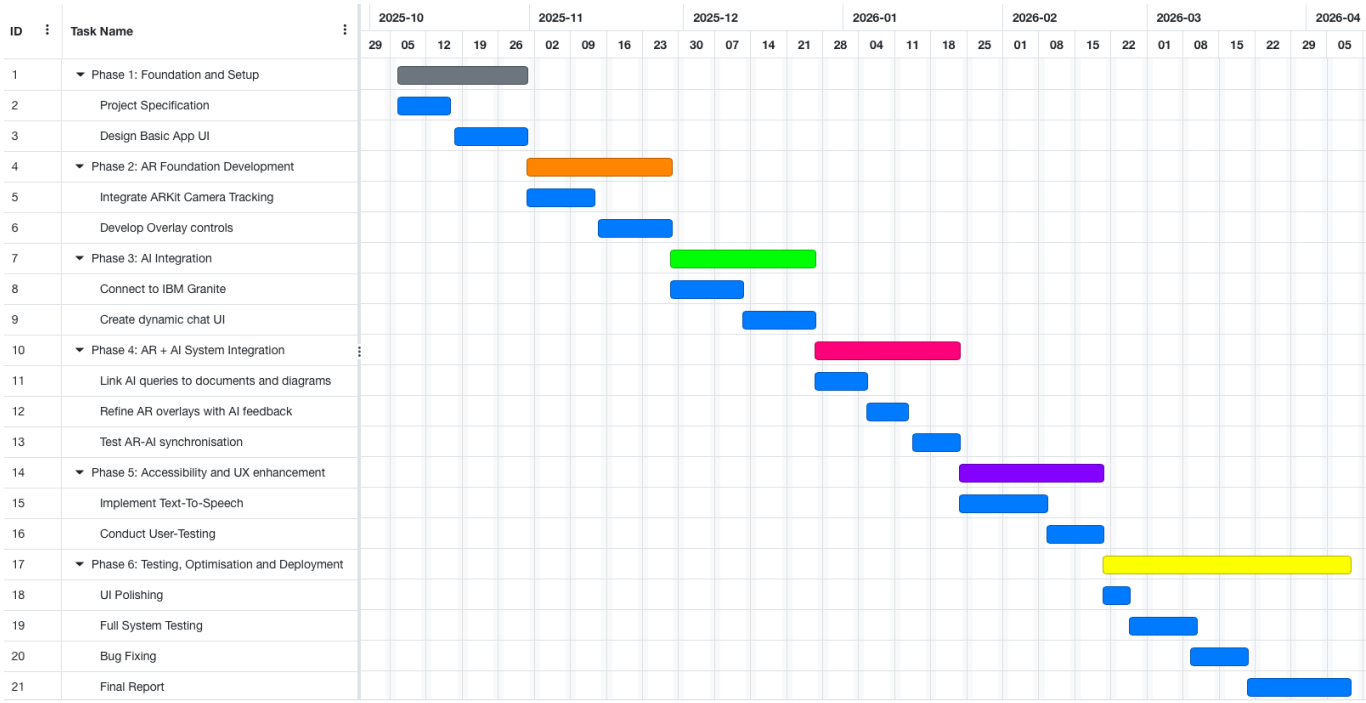


Figure 2: Gantt Chart of Planned Work Schedule

## 5.3 Resource Management

The project will be primarily developed by a single developer, with support and guidance from the assessors. The developer will be responsible for all aspects of the project, including design, implementation, testing and documentation. Access to the essential software tools and platforms, such as IBM Granite and IBM Cloud, will be provided by IBM. Other resources, such as hardware for testing, necessary software and other materials, will be handled by the developer. Except for the resources provided by IBM, the developer will be responsible for acquiring any additional resources needed for the successful completion of the project. In the case of any type of resource unavailability, such as laptop malfunction or equipment failure, the developer will rely on the university's resources provided to students.

## 5.4 Risk Management

The following risks have been identified for the project, along with their descriptions, risk levels, likelihoods and mitigation strategies.

### 5.4.1 Technology Limitations

- Risk description: There is unfamiliarity with some tools, libraries, or frameworks, which may cause delays or reduced performance.
- Risk Level: Tolerable
- Risk Likelihood: Moderate

- Mitigation Strategy: Conduct thorough research and allocate time for learning and experimentation with new technologies.

#### 5.4.2 Rollback Challenges

- Risk description: Lack of a version control system could prevent from rolling back to the software's last stable state in case of errors
- Risk Level: Catastrophic
- Risk Likelihood: Low
- Mitigation Strategy: Utilise github to always maintain a stable version of the software and updating it when being sure that the changes will not affect its usability.

#### 5.4.3 Testing Risks

- Risk description: Insufficient testing may reduce confidence in the software
- Risk Level: Serious
- Risk Likelihood: Moderate
- Mitigation Strategy: Unit tests will be designed to test the software to make sure that it is working properly and user testing will be conducted to gather feedback and identify any issues.

#### 5.4.4 Time Management

- Risk description: Underestimating task duration or improper prioritization might result in delayed work.
- Risk Level: Serious
- Risk Likelihood: Low
- Mitigation Strategy: Meetings with the assessors will be held weekly to ensure that the project is on track and any issues are addressed promptly.

#### 5.4.5 Requirement Misalignment

- Risk description: During the development of the software, the end product might not be the same as the one described in the deliverables due to unforeseen circumstances.
- Risk Level: Catastrophic
- Risk Likelihood: Low
- Mitigation Strategy: Regular meetings with the assessors will be held to ensure that the project is on track and any issues are addressed promptly.

#### 5.4.6 Underestimating Sprint Workload

- Risk description: Tasks may take longer than expected due to AI and AR complexities, which may lead to incomplete sprints and delays.
- Risk Level: Serious
- Risk Likelihood: Moderate
- Mitigation Strategy: Buffer time will be allocated in each sprint to accommodate unforeseen challenges, and tasks will be prioritized to ensure critical features are completed first.

# 6 Conclusion

Summarize the key points of the specification document and outline next steps.