Software Requirements Specification

Project Title: UI/UX AUTOMATION USING LLM

Project Code:

Project Advisor:

Dr. Fahad Maqbool

Project Manager:

Dr. Muhammad Ilyas

Project Team:

- Muhammad Dawood (BSCS51F21S089) Team Leader
- Muhammad Rashid (BSCS51F21S084)
- Ghulam Rasool (BSCS51F21S080)

Submission Date:

11/26/2024

_____ _____

Signature Project Manager's

Document Information

Category	Information
Customer	UI/UX Designers, Design Teams, and Developers.
Project	UI/UX Automation using LLM
Document	Requirement Specifications
Document Version	1.0
Identifier	PGBH01-2003-RS
Status	Draft
Author(s)	Muhammad Dawood, Muhammad Rashid, Ghulam Rasool
Approver(s)	Prof. Dr. Muhammad Ilyas
Issue Date	Sept. 10, 2024
Document Location	
Distribution	1. Dr. Fahad Maqbool 2. Dr. Muhammad Ilyas 3. University of Sargodha

Definition of Terms, Acronyms and Abbreviations

Term	Description
LLM	Large Language Model, used for generating designs and contextual suggestions.
GraphRAG	Graph-based Retrieval-Augmented Generation, uses knowledge graphs for context-aware LLM responses.
OWL	Web Ontology Language, defines ontologies for semantic consistency.
Neo4j	Graph database used to store and query UI/UX design knowledge.
UI	User Interface, the visual part of the system where users interact.
UX	User Experience, the overall experience a user has with a product or service.
Figma	A collaborative interface design tool used for high-fidelity design

Table of Content

1. Introduction

1.1 Purpose of Document

This document provides the software requirements for the program Automating UI/UX Design with LLMs. It serves as a specification for system developers, advisors and stakeholders, describing the system's what it can and cannot do features.

1.2 Project Overview

A remarkable aspect of the user interface in the contemporary world is the use of Large Language Models in conjunction with a Neo4j knowledge graph and OWL ontologies. GraphRAG is how it is termed, and what it is to do is to provide relevant automation of wire frame creation, automatic design validation, and high-fidelity Outputs in Figma.

1.3 Scope

The system will:

• Automate the creation of wireframes and validations regarding the created designs.

• Retrieval of the external user interface or user experience rules and parts with the help of GraphRAG.

• Transferring of designs into Figma.

Out of scope:

• Addressing very specific and domain complex UI/UX requirements which exceed the ontological definition.

The production of comprehensive interactive models.

2. Overall System Description

2.1 User Characteristics

- Primary Users: UI/UX designers, developers.
- Secondary Users: Project managers, clients for reviewing designs.
- Users will have basic knowledge of design principles and Figma.

2.2 Operating Environment

- Hardware: Standard PC or laptop with internet access.

Software:

- Neo4j database for knowledge graph.

- Protégé for OWL ontology development.
- GraphRAG integrated with LLM APIs (e.g., OpenAI GPT).
- Figma for exporting designs.

2.3 System Constraints

- Dependence on internet connectivity for LLM queries.
- Knowledge graph scalability may be limited by Neo4j configurations.
- Ontology updates require manual effort.

3. External Interface Requirements

3.1 Hardware Interfaces

- Standard computing devices (laptops/desktops) with minimum 8GB RAM and 256GB storage.

3.2 Software Interfaces

- Neo4j 5.0 or later.
- OpenAI API for LLM integration.
- Protégé for ontology management.
- Figma API for design export.

3.3 Communication Interfaces

- REST API for communication between LLM and GraphRAG.
- Secure HTTPS protocols for data transmission.

4. Functional Requirements

4.1 Wireframe Generation

- Description: The system should automatically create wireframes from users written instructions like “Create a page with a sidebar menu and two graphs”.
- Trigger: The submission of design request or description by user.
- Expected Output: The system creates a wireframe which includes the design details which were needed and it is generated as requested.
- Business Process: The system integrates GraphRAG, in querying the knowledge graph and LLM to create necessary wireframe components

4.2 Design Validation

- Description: the system shall check or validate the automatically generated wireframes against UI/UX principles which are known and defined in the system ontology.
- Trigger: the action of amending or uploading a wireframe and validation process began.
- Expected Output: The result of the wireframe in question – an automated report that confirms or deny that wireframe designed is in accordance with the specifications provided in the document in form of wireframes.
- Business Process: The wire frame elements are validated in consideration of the OWL defined rules and any feedback applicable to the situation is provided together with suggestions for improvement.

4.3 Design Feedback and Suggestions

- Description: When a user provides an input, based on intelligent analysis, the system generates objective improvements of the design.
- Trigger: When the user once the wireframe is generated, requests feedback or suggestions regarding the fully generated wireframe they have no idea about how it looks like all present in the wireframe.
- Expected Output: The resulting output of the system will include some suggestions such as placing more space in some areas, changing colour or shifting components closer together or apart.

4.4 Export to Figma

- Description: Users are permitted to import the created wireframes in higher fidelity as finished products to Figma App.
- Trigger: When satisfied with the selected wire frame, the user exports.
- Expected Output: The wire frame is altered to a Figma-compatible file format and is ready for transfer or loading directly into Figma.
- Business Process: To foster integration with Figma, the system changes the format of the wireframe to Figma's design format through Figma's API.

4.5 Knowledge Graph Management

- Description: The system should be responsible for the creation and nourishment of UI/UX knowledge graphs which contain design patterns, principles, among other relevant information.
- Trigger: When administrators or developers add new design patterns, ontologies, or modify existing ones in the knowledge graph system.
- Expected Output: Raise the knowledge graph by adding new design patterns, principles and make corrections so as to have an up to date knowledge base in the system.
- Business Process: System administrator privileges are employed to carry out modifications to knowledge graph so that the proper knowledge is retained.

5. Non-functional Requirements

5.1 Performance Requirements

- Speed: Execute wireframe generation and validate designs in less than 5 seconds per query.
- Precision: The UI/UX principles in the outputs must match with that defined in the OWL ontology with precisions rate by 95%.
- Capacity: The system shall support 1,000 concurrent users, scalable to 10,000 users.
- Reliability: The service must be 99.9% reliable.

5.2 Safety Requirements

- Things you can do to avoid data loss due to hardware/software failures is to make periodic backups of the knowledge graph and ontology files.
- Safeguard from unauthorized deletions or modifications of important design components stored in the database.
- Make sure there are fallback mechanisms for generating wireframes when the LLM API goes down and the outage is recovered.

5.3 Security Requirements

- All user data and communications should be encrypted using AES-256 standards.
- Restricting unauthorized modifications to the ontology and knowledge graph by implementing role-based access control.
- Secure user login to the system using OAuth 2.0 with multi-factor authentication.
- In all cases where it is applicable, relevant data protection regulations; ensure GDPR compliance.

5.4 Business Rules

- UI/UX Designers: Generate wireframes and validate designs plus everything else done in Figma exports.
- Project Managers: View progress reports plus validate designs and approve exports.
- System Administrators: Update the ontology plus manage the knowledge graph and everything about maintaining LLM API integrations.
- Only certain registered users are granted access to the system features; this ensures accountability and the ability to delineate between different role-based functions within the system.

5.5 User Documentation

1. **User Manual:**

- A guide outlining the procedure to follow in navigating the system, generating wireframes and exporting designs to Figma.

2. **Online Help:**

- Built-in support that will provide a basic explanation of UI/UX principles and functionalities of the system.

3. **Tutorials:**

- Video and text tutorials on using the system for wireframe generation and design validation.

6 Assumptions and Dependencies

6.1 Assumptions

1. **Availability of LLM APIs:**

- It is assumed that the LLM APIs to be used during the project life cycle— for example, OpenAI GPT-4—will be available and stable. Any unavailability or major changes in such services might impact the system to appropriately handle the user

input and generate designs.

2. **Knowledge Graph and Ontology Data:**

- The system will assume that the knowledge graph, constructed using Neo4j, will be accurate and scalable. Any changes related to the structure of the graph may lead to incorrect design suggestions or errors in wireframe generation due to inconsistencies in data.

3. **User Accessibility:**

- The system assumes that the users (designers, developers, and project managers) will have access to a stable internet connection for querying the LLM, accessing the knowledge graph, and using the Figma API. In rural or low-connectivity areas, this may affect the system's performance.

4. **Standardization of Design Patterns:**

- It is expected that the UI/UX design principles set in the ontology shall be generally applicable to a wide range of projects. Changes in industrial standards or the adoption of new design trends may require the knowledge base to be updated.

5. **Security Policies and Compliance:**

- It is assumed that the users will adhere to the established authentication and authorization protocols (OAuth2 with multi-factor authentication). Any changes in the security requirements or any new policy, such as new compliance regulations, may need more changes in the system.

6. **Platform Compatibility:**

- It is assumed that this system is fully compatible with major web browsers— Chrome, Firefox, Edge—and operating systems—Windows, macOS. A significant change in these platforms or the introduction of a new technology may necessitate an additional check for compatibility.

7. **Feedback Mechanism:**

- The functionalities for wireframe generation and design validation input are assumed to be provided by the users on a regular basis, which in turn will help improve the recommendations of the system and the general user experience.

6.2 Dependencies

1. **LLM API Availability**

- The system relies on external application programming interfaces (APIs) such as OpenAI GPT-4 for natural language processing, and for wireframe creation. Any change that may be observable in the services offered in the API services, price or restriction may alter the prospect of the system.• Exporting wireframes with this system employs the Figma API.

2. Figma API Integration

- The system depends on the Figma API for exporting wireframes. Some impacts of changes are: changes in Figma API, including rate limits of requests, new features, or deprecations of features, changes to the terms of service.

3. Knowledge Graph Management Tools

- Neo4j is used for the management of the knowledge graph. Neo4j platform or its capabilities level changes (e.g., version upgrades, security vulnerabilities) may impact the graph's operation and scalability.

4. Cloud Infrastructure (Hosting Services)

- The hosting, model inference and API integration is done through cloud infrastructure (e.g., Google Cloud Platform). Smooth operations are highly dependent on these services being used and at what cost and availability.

5. User Training and Documentation

- The availability of training materials and user documentation is another critical part of the system's success. Many times, if the users are not sufficiently trained or informed, it may result in improper use or under- utilization of features.

6. External Stakeholders' Availability

- The project relies on prompt feedback and participation from external stakeholders (e.g., project managers, designers) to confirm features and make sure the system satisfies business requirements."

7. Data for Ontology Development

- The development of the ontology depends on finding the right UI/UX design principles and business models. If this information is missing or changes over time, the ontology needs to be updated.

8. References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
PGBH01-2003-Proposal	Project Proposal	Oct 20, 2003	<Give the path of your Project repository/Folder>
RAG2023	GraphRAG Explained: Enhancing RAG with Knowledge Graphs	2023	Medium (https://medium.com/@zilliz_learn/graphrag-explained-enhancing-rag-with-knowledge-graphs-3312065f99e1)
OWL2004	OWL Reference	Feb 10, 2004	W3C (https://www.w3.org/TR/owl-ref/)
AIUX2023	Artificial Intelligence (AI) for User Experience (UX) Design	Aug 2023	Information Technology and People Journal
MFG2024	Making Knowledge Graphs Work for Smart Manufacturing	2024	Journal of Manufacturing Systems
ADDSYS2020	Machine Learning and Knowledge Graph-Based Design Rule Construction	2020	Additive Manufacturing Journal
UIMOBILE2022	UI Design Patterns and Ontology Models for Adaptive Mobile Applications	2022	Personal and Ubiquitous Computing Journal
PROTEGE1999	Protégé: A Free, Open-Source Ontology Editor and Framework for Building Intelligent Systems	Nov 11, 1999	wikipedia: http://protege.stanford.edu/

9. Appendices

Appendix A: Glossary

Term	Description
LLM	Large Language Model, used for generating designs and contextual suggestions.
GraphRAG	Graph-based Retrieval-Augmented Generation, uses structured knowledge graphs for context-aware responses.
Ontology	A structured framework for organizing information, defining relationships between concepts.
Protégé	An open-source ontology editor used to create OWL-based ontologies.

Neo4j	A graph database used to store and query structured data for UI/UX design.
Figma	A collaborative design tool used for exporting high-fidelity wireframes.

Appendix B: Workflow Diagram

Input Design Requirements:
The user enters the design description into the system.

Process Input Using LLM:
LLM process input and query information Neo4j knowledge to obtain key points and definitions.

Generate Wireframe:
The system creates wireframes based on the collected data and analyzes them against the ontology.

Validation and Feedback:
Validates and provides input to the resulting wireframes against predefined UI/UX principles store d in the ontology.

Export to Figma:
The final wireframe was imported into Figma for further development and use.