**SE 4485: Software Engineering Projects**

Spring 2024

**Detailed Design Documentation**

|  |  |
| --- | --- |
| Group Number | 10 |
| Project Title | Internet Research Assistant |
| Sponsoring Company | The Fellows Consulting Group |
| Sponsor | Jeff |
| Students | Bakr Alkayali  Chloe Pascual  Vi Le  Ikraam Rahman  Mohammad Chauhan |

ABSTRACT

The detailed design document is a comprehensive outline of the design of our internet research assistant. It illustrates the ideal visual UI of the website and focuses on how the interface interacts with the user. Moreover, the static model class diagrams and dynamic model sequence diagram depicted also show all the other design considerations such as the relationships and interactions between the AI tools, database, logger, etc and the previously mentioned user facing interface. The GUI presented is a Figma mockup which will be translated to the frontend of the project implementation to the extent possible. In the latter half the application requirements finalized from previous stages of project inception are mapped to the detailed design choices made to make the application implementation and delivery successful as it would fulfill requirement criterion.

TABLE OF CONTENTS

LIST OF FIGURES ...................................................................................................................................2

LIST OF TABLES .....................................................................................................................................2

INTRODUCTION .....................................................................................................................................2

GUI DESIGN .............................................................................................................................................3

LIST OF FIGURES

Figure 1 – Graphical User Interface .......................................................................................................... 3

Figure 2 – Class Diagram .......................................................................................................................... 4

Figure 3 – Sequence Diagram ..................................................................................................................... 5

LIST OF TABLES

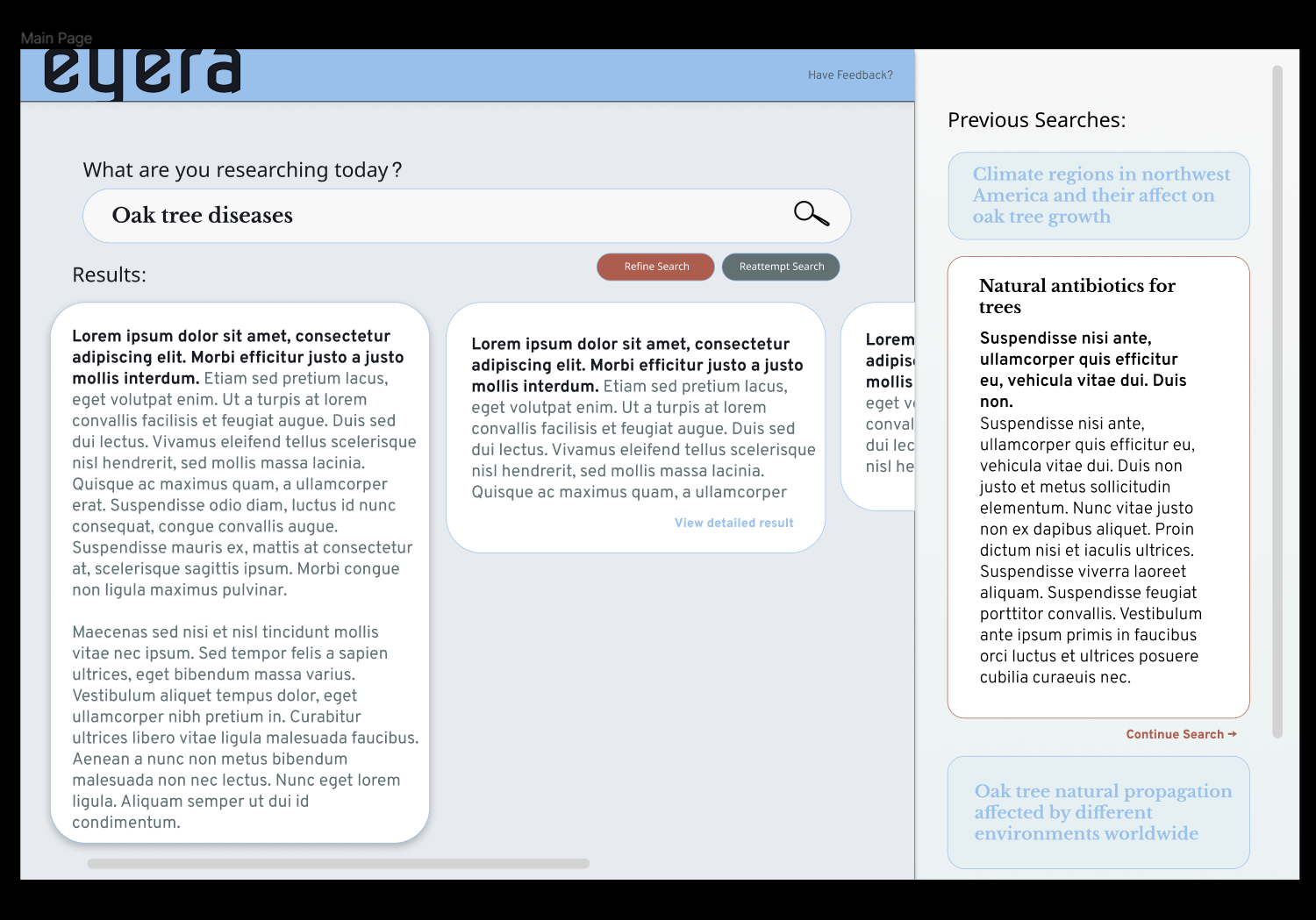
Table I – Requirements to Detailed Desgn Model Traceability Matrix .....................................................6-8

INTRODUCTION

* The detailed design documentation provides a complete description of the system and how it was developed. This document mostly goes into the intricacies of the system’s architecture, functionality, and underlying technologies.
* The document’s main purpose is to showcase how we created the design for our internet research assistant. It goes into depth how we visualized the UI for the website and how each function works that the user will interact with when using the system. The diagrams shown depict the relationship between AI tools, database, logger, etc and the previously mentioned user facing interface.
* The main user interface is built using a GUI which is then presented through a Figma mockup. The user interface asks the user what they’d like to search which then displays the search results. A side menu will display previous search results to the user.
* The structure of the document starts with an introduction to the document's purpose and scope which is then followed by the diagrams that show off how we created the user interface and our chosen architecture. The comprehensive traceability matrix which maps the functional and non-functional requirements to the corresponding architectural components. The documents end with a clarification on the configuration management and the engineering standards in use.

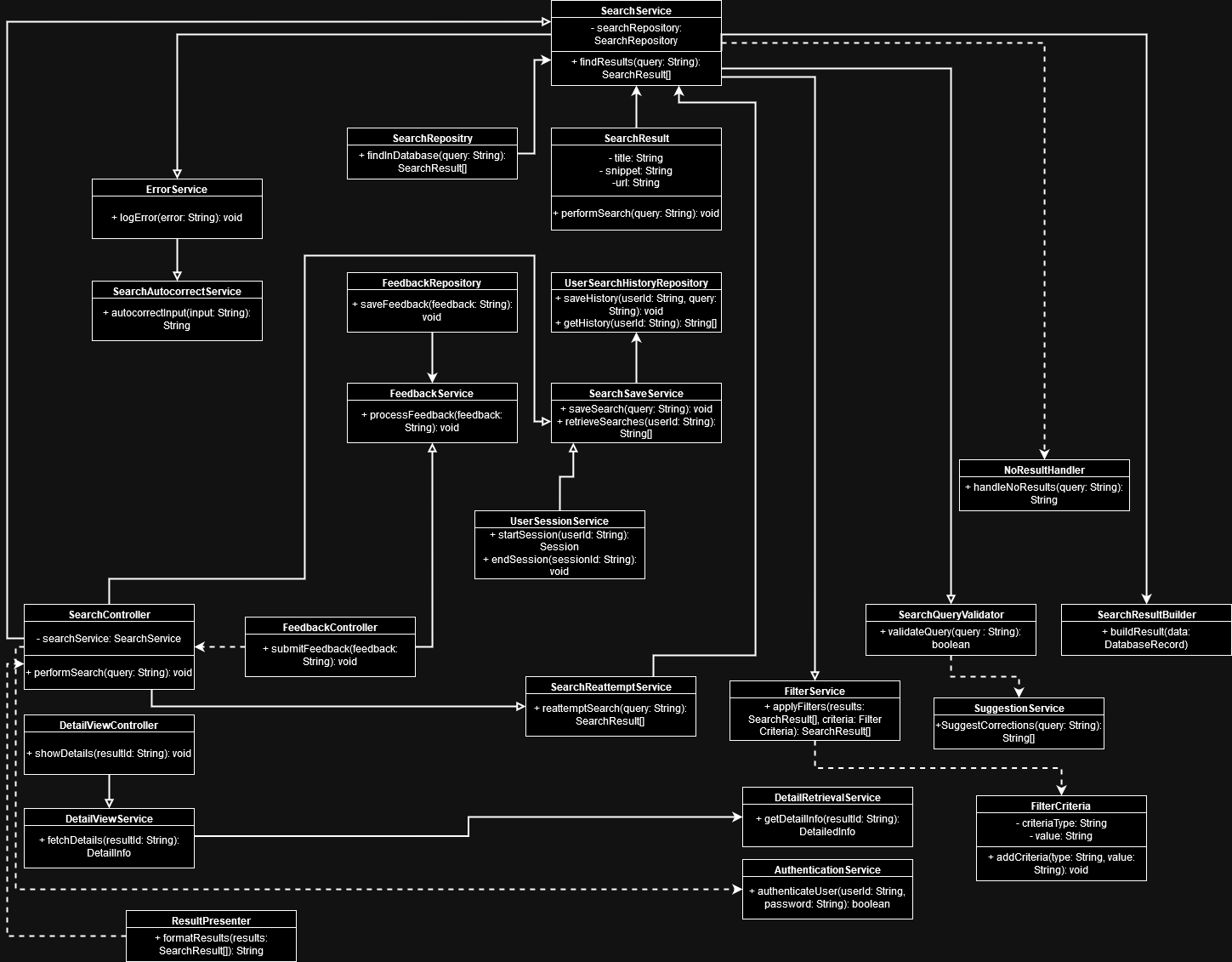
GUI (Graphical User Interface) Design

**Figure 1**

Fig 1. Main Graphical User Interface of Eyera

STATIC MODEL CLASS DIAGRAMS

**Figure 2**

 Fig. 2. Static Model Class

DYNAMIC MODEL SEQUENCE DIAGRAMS

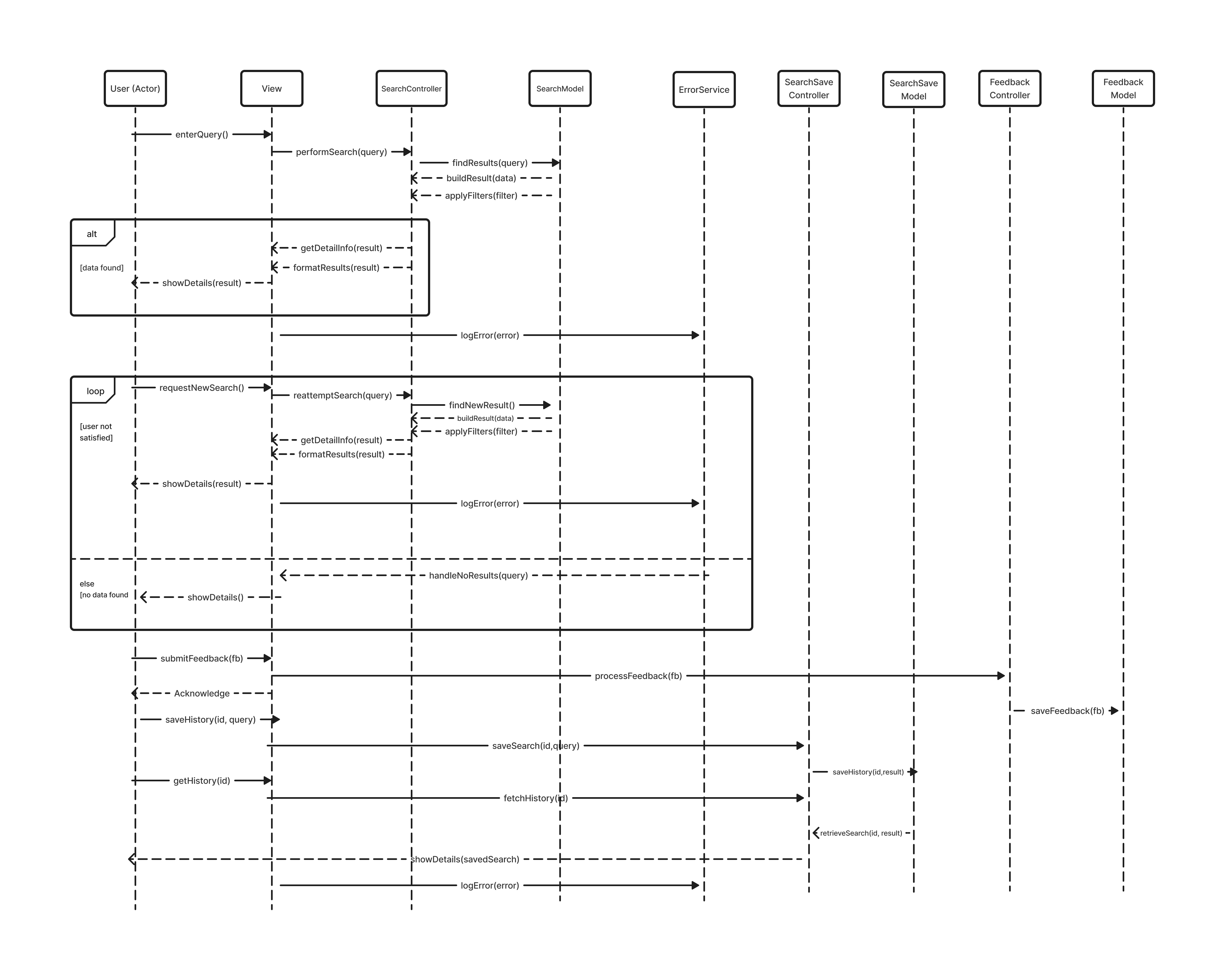


Fig. 3. Sequence Model Diagram

RATIONALE FOR YOUR DETAILED DESIGN MODEL

Our detailed design model covers both dynamic and static versions of the software with a class diagram and a sequential diagram. This provides a full view of the digital architecture in application and action that will help with the development process. These diagrams consider the found requirements and establish classes, objects, and methods depicted in the class diagram and show how these things communicate in the sequential diagram. With this detailed design model, the architectural factors that determine how requirements will be met have been fully identified and documented.

TRACEABILITY FROM REQUIREMENTS TO DETAILED DESIGN MODEL

TABLE 1

Requirements to Detailed Design Model Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Design Element | Design Description |
| FR1 | Perform Search/Enter Search Query | 1. SearchController 2. SearchService 3. SearchQueryValidator 4. SearchResultBuilder 5. SearchRepository 6. SearchResult | 1. Handles UI requests to initiate a search. 2. Implements the logic to process and execute a search. 3. Validates the user input against search criteria. 4. Constructs the search results to be returned to the UI. 5. Interacts with the database to fetch search results. 6. Represents the data of an individual search result |
| FR2 | Validate Query | 1. SearchQueryValidator  2. SuggestionService | 1. A dedicated class or service that checks the search string's validity.  2. Provides suggestions or corrections for potentially invalid queries. |
| FR3 | Display Search Results | 1. ResultPresenter  2. SearchResult | 1. A component within the View that takes search results and formats them for display.  2. A data class representing individual search results |
| FR4 | Handle No Results Found | 1. NoResultHandler | 1. Generates a user-friendly message or alternatives when no results are found for a search query. |
| FR5 | Refine Search Results / Apply Filter | 1. FilterService  2. FilterCriteria | 1.Applies various filters to the search results based on user selection.  2. Holds the filtering parameters (criteria) that FilterService uses. |
| FR6 | View Search Result Details / Select Search Result | 1. DetailViewController  2. DetailViewService | 1. Controls the detailed view of search results.  2. Fetches additional information for a detailed view. |
| FR7 | Display Detailed Information | 1. DetailRetrievalService | 1. Retrieves detailed information about a particular search result from the database. |
| FR8 | Save and Retrieve Saved Searches | 1. SearchSaveService  2.UserSearchHistoryRepository | 1. Manages saving and retrieving searches for users.  2. Interfaces with storage to save and retrieve user search history. |
| FR9 | Reattempt Search for Updated Results | 1. SearchReattemptService | 1. Handles logic for re-executing past searches to fetch updated results. |
| FR10 | User Feedback Submission | 1. FeedbackController  2. FeedbackService  3. FeedbackRepository | 1. Accepts feedback from the user through the UI  2. Processes and persists feedback to the database.  3. Handles storage and retrieval of user feedback. |

NFRs

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Design Element | Design Description |
| NFR 1 | Usability-User Interface | 1. SearchController 2. DetailViewController 3. DetailViewService 4. Result Presenter 5. SearchService | 1. Display user interface 2. Controls the detailed view of search results. 3. Fetches additional information for a detailed view. 4. Component within the View that takes search results and formats them for display. 5. Implements the logic to process and execute a search. |
| NFR 2 | Usability-History & Navigation | 1. UserSessionService 2. SearchSaveService 3. UserSearchHistoryRepository | 1. Starts the user search session 2. Saves the user’s search results to the repository 3. Repository holds previous search results that the user has searched |
| NFR 3 | Performance-Response Time | 1. SearchService 2. SearchRepository | 1. Starts the search service for the user 2. Keeps search results in repository in case user would like to revisit previous searches |
| NFR 4 | Performance-Scalability | 1. SearchRepository | 1. Keeps search results in repository in case user would like to revisit previous searches |
| NFR 5 | Reliability | 1. ErrorService 2. NoResultHandler | 1. Show the user an error if system crashes 2. Show the user an error if no results appear |
| NFR 6 | Efficiency | 1. UserSearchHistoryRepository | 1. Allows the user to quickly go and view previous searched topics |
| NFR 7 | Accessibility | 1. FeedbackService | 1. Gives the user the option to provide feedback to the user |

EVIDENCE THE DESIGN MODEL HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

1. Name of the CM tool: GitHub
2. Version number of before: [7ab01ed](https://github.com/Bakr8724/CapstoneSearchTool/commit/7ab01ed2bddfd1c102b4ca3cd2830ffac9b54ac0)
3. Version number after:
4. Difference between the two: As stated below, additional information were added. Along with correction of model diagrams.
5. Review of each change:
   1. Before: Added Standards, References, formatted document outline, added GUI, added partial traceability table, added dynamic model diagram.
   2. After: Added Traceability matrix table info, added static diagram model, revised statid diagram model, added remaining information need.
6. Other info:

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

* IEEE Std 1016-1998-(Revision-2009): Software Design
* ISO/IEC 12207:2017: Systems and software engineering – Software life cycle processes.
* ISO/IEC 15288:2023: Systems and software engineering – System life cycle process.
* ISO/IEC 29148:2018: Systems and software engineering – Life cycle processes – Requirements engineering.
* IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.

ADDITIONAL REFERENCES

* Boehm, B., 1988: A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes.
* Cockburn, A., 2000: Writing Effective Use Cases. Addison-Wesley.
* Sommerville, l., 2015: Software Engineering. 10th Edition, Addison-Wesley.
* Martin, R.C., 2003: Agile Software Deveolpment: Principles, Patterns, and Practices. PRentice Hall.
* Gamma, E., Helm, R., Johnson. R., Vlissides, J., 1994: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
* Larman, C., 2012. Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development. Pearson Education
* Hyman, B., 1998. Fundamentals of Engineering Design. New Jersey: Prentice Hall
* Simon, H.A., 2014. A Student's Introduction to Engineering Design: Pergamon Unified Engineering Series (Vol. 21). Elsevier