SYSTEM MAP

*This system map illustrates the key screens and interactions of the InQuery Documentation Assistant. The implementation illustrated here is envisioned as an augmentation to M*Modal's existing product, Fluency Direct.*

WORKFLOW SCENARIOS

- **Scenario 1:** Immediate Query Response (p. 25)
- **Scenario 2:** Deferred Query Response (pg. 33)
- **Scenario 3:** Externally Generated Query (pg. 38)

new query is triggered

click on a query in the inbox

click 'Go to Query' button

click 'Query Inbox' button

click 'New Note' button

click 'Create Note' button

click query in list

click query button in text

confirm changes in query dialog

click 'Sign Note' button

click 'Sign Note'

dictate deficient note

▲ New Query (p. 49)

A notification is displayed below the Pill for new queries not triggered during note authoring. It provides a summary and access to the query.

▲ Query Dialogs (p. 50)

Queries are communicated through Query Dialogs. These dialogs pop-up in the context of a potential deficiency. The dialogs both explain the suspected deficiency and provide the physician with actions to take in order to address the deficiency. Any changes made via the Query Dialogs are immediately reflected in the physician's note.

EMR

▲ The Pill (p. 42)

The Pill is a minimal window that stays open in the foreground while a physician works with their EMR. It is closely based on the existing interface for Fluency Direct. In our implementation of InQuery, the Pill becomes the access point for the Note Editor and Query Inbox. The Pill also displays a count for the number of unanswered queries.

▲ Query Inbox (p. 48)

The Query Inbox displays all unanswered queries assigned to a physician. It displays summary information about each query.

▲ The Note Editor (p. 44)

The Note Editor provides an environment for a physician to dictate or type a new note. Our implementation of the Note Editor is not fully developed but provides a proof of concept for how InQuery would be implemented in a note authoring tool.

▲ Query Buttons (p. 45)

As a physician creates their note they are alerted to potential deficiencies with in-context Query Buttons. Terms in their note that are associated with a deficiency are highlighted by placing the term on a button within the text. These buttons provide access to queries regarding the deficiency. The queries are also aggregated in a list on the left for easy access when dealing with long notes.

▲ Query Response (p. 47)

In each Query Dialog physicians are given the choice to either update their note based on their selection or to dismiss the query. If they choose to update the text, the text on the query button is replaced based on their selection in the dialog. Otherwise, the original text is restored. In either case, the text is highlighted so that the physician can clearly see the outcome of their selection. The highlighted text can also be right-clicked to return to the query dialog.

▲ Pushing Note to the EMR

In our implementation, the Note Editor is a part of Fluency Direct, separate from the physician's EMR. Upon signing the note the text would be pushed back into the EMR.

EMR

▲ Entering MRN (p. 43)

This intermediate step asks the physician to enter a patient's medical record number in order to sync the new note with the EMR.

▲ Adding to Query Inbox

When a note is signed while it still contains unanswered queries, the queries are added to the Query Inbox. Queries can then be addressed later.