Open Source Engine Integration Software Design Document

Harris SmartWorks
Open Source Engine Integration
Computer Science Dept
COSC470
Okanagan College

Aubrey Nickerson
Bao Mai
Derek Manchee
Joesph Egely
Keaton Canuel

December 4, 2020

Revision Sheet			
Revision	Date	Summary of Changes	Name
Version 1.1	11-8-2020	Baseline Draft	Derek Manchee
Version 1.2	11-9-200	Update of Architecture and Use Case diagrams, as well as clarification of text descriptions.	Joesph Egely
Version 1.3	11-16-2020	Added Images for GUI	Aubrey Nickerson
Version 1.4	11-20-2020	Updated all diagrams	Joesph Egely

Contents

1 Introduction

1.1 Purpose

To develop an HL7 listener, parsing, and storing messages in a database for future analysis.

1.2 Scope

Software to be developed:

- A listener to catch HL7
- Sorting program to store messages into appropriate JSON objects
- Database for storing messages
- Storing JSON objects appropriately in database

The listener will catch HL7 messages sent across the network, and send a copy into the system for sorting and storage, the original continuing to the recipient. Once in the system, messages are sorted by event type, put into appropriate JSON objects, and stored in the database.

The system is written to collate and store messages system-wide to improve tracking of communications among staff. In addition, message content will be used to insert, update, and remove patients from the database. The system must interact with the existing communication network, which the listener rests on.

This document will receive updates throughout the development process to describe subsystems and architecture, as well as provide relevant information for future development and maintenance.

1.3 Definitions, Acronyms, and Abbreviations

FHIR- Fast Healthcare Interoperability Resource, is a framework for organizing data that will be stored. HL7 - Health Level 7, a messaging standard used in healthcare. GUI - Graphical User Interface. BI - Business Intelligence

2 Design Overview

2.1 Background Information

Stakeholders in this project are Harris Healthcare Group and the health institutions they work with. This project is very low risk, as the system being developed is fairly independent from existing systems. Our system will provide a solution to sorting, parsing, and storing HL7 messages.

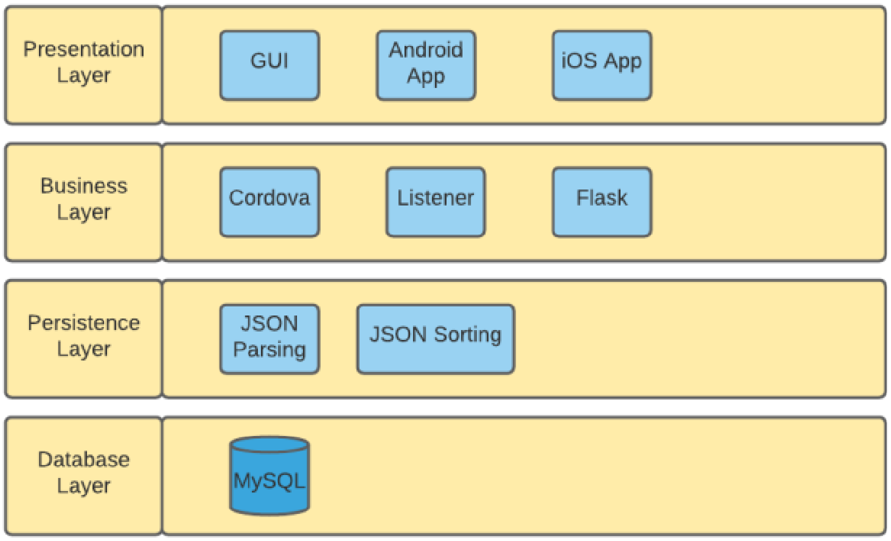
2.2 Alternatives

No alternatives have been considered, as the system requirements are well laid out and only one method is known to us to obtain the data needed to meet them.

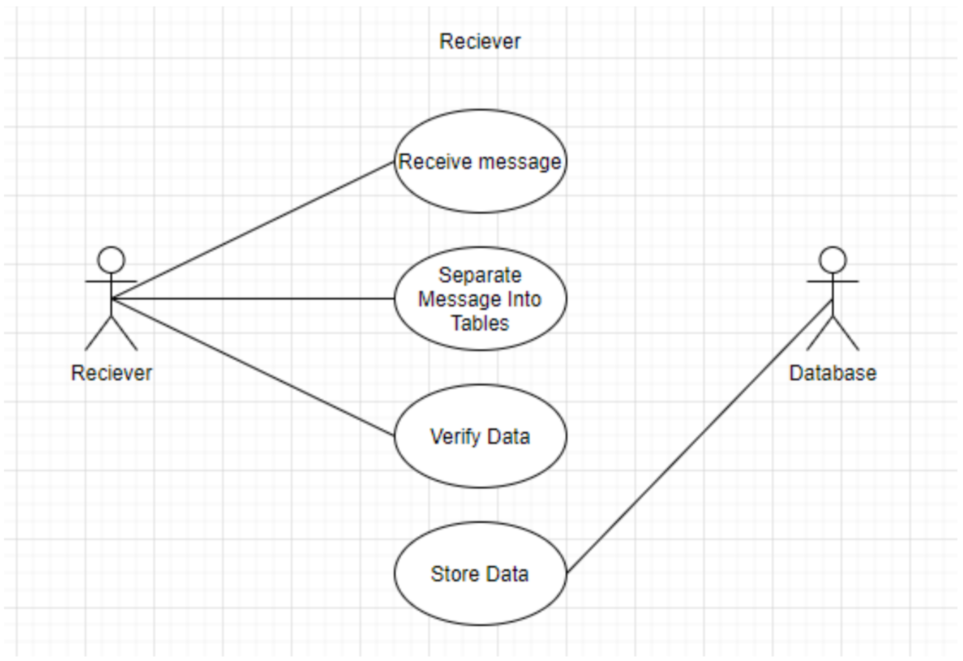
3 User Characteristics

System users will be IT teams in relevant institutions and healthcare professionals using the HL7 communications standard. Health professionals will need little to no expertise in their interactions, as the system as they interact with it will be automated or operated with a simple GUI. IT teams will need to be competent in Python programming and database systems. The system design meets user requirements by sorting, parsing, storing, and operating on messages in real time.

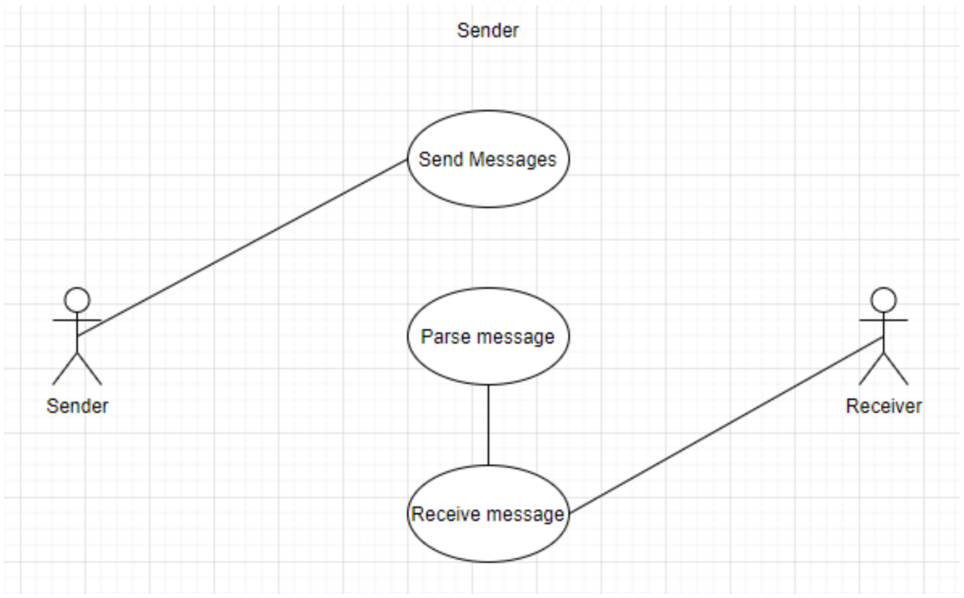
4 System Architecture



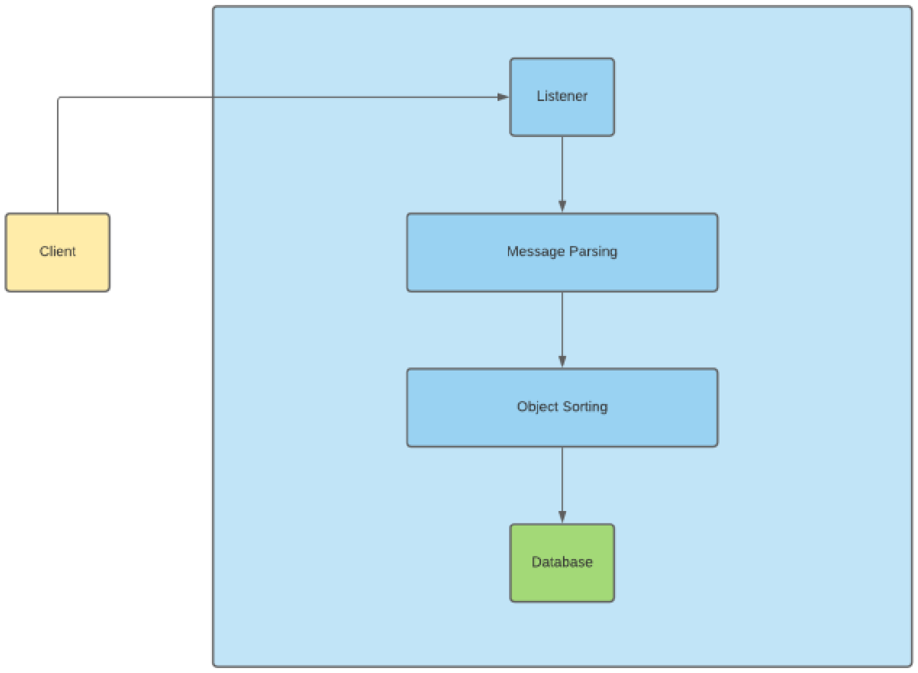
The system is built in four layers: Presentation, Business, Persistence, and Database. On the Presentation layer, the one a user generally interacts with, we implement our web-based GUI and iOS/Android apps. On the Business layer, we support the Presentation Layer with Cordova - a tool for developing apps from HTML and CSS - our custom listener, and Flask - a Python library for full stack web development. The Persistence layer takes data from the Business layer and prepares it for storage, by sorting it into JSON objects by event type, and parsing them from there into a format suitable for storage. The Database layer consists of a MySQL database.



This Use Case describes the functioning of the receiver and data validation. The listener spends the majority of its time waiting for data travelling across the network. When data is seen, the listener catches data travelling across the network, and sends it for processing. During processing, it is validated as a proper HL7 message and sorted into JSON objects. These objects are then split apart to be stored in the proper tables and rows.



This Use Case describes database and sender functioning. Upon receiving data from the receiver, it is processed appropriately to be stored into rows and tables as per schema design before being input into the tables. Alternatively, data can be retrieved and processed for various business needs such as analysis.



This use case describes the overall data flow of the system. A client sending a message across the network has that message caught by the listener, which sends it for parsing and sorting before being stored in the database.

images/OEI$_{A}ctivity_{D}iagram_{1}.vpd.png$

This diagram depicts the data flow for a message through the system. On being caught by the listener the HL7 message is scanned, if it is an Admission, Discharge or Transfer (ADT) message it is converted to an FHIR format and sent to the FHIR server. If it is not of those types, or the heading is missing or incorrect, the message processing halts before format changes take place. Here, it is stored, and displayed on the server-side GUI when retrieved.

5 Detailed Design

5.1 Description for Listener

5.1.1 Processing for Listener

Data is caught during standard traffic through the network, and sent to the next component in the system, the sorting system.

5.1.2 Component Listener Description

The listener is a software interface on the network that is used to collect data for the rest of the system.

5.2 Description for Sorting System

5.2.1 Processing for Sorting System

Data sent from the listener is scanned and sorted into a relevant JSON object according to its event type. Currently supported events are A01, A02, and A08, representing patient admittance, patient discharge, and updates on patient records respectively.

5.2.2 Component Sorting System Description

The Sorting System works to prepare message data for parsing, action, and storage in the database by the next component, the parsing system.

5.3 Description for Parsing System

5.3.1 Processing for Parsing System

The parsing system intakes a JSON object sent from the Sorting System, inserts, deletes, or updates data as relevant to the message type, and stores the message for the purposes of record keeping and potential analysis.

5.3.2 Component Parsing System Description

The Parsing System serves to process message data as needed and store it in the database.

6 Interface Requirements

6.1 Required Interfaces

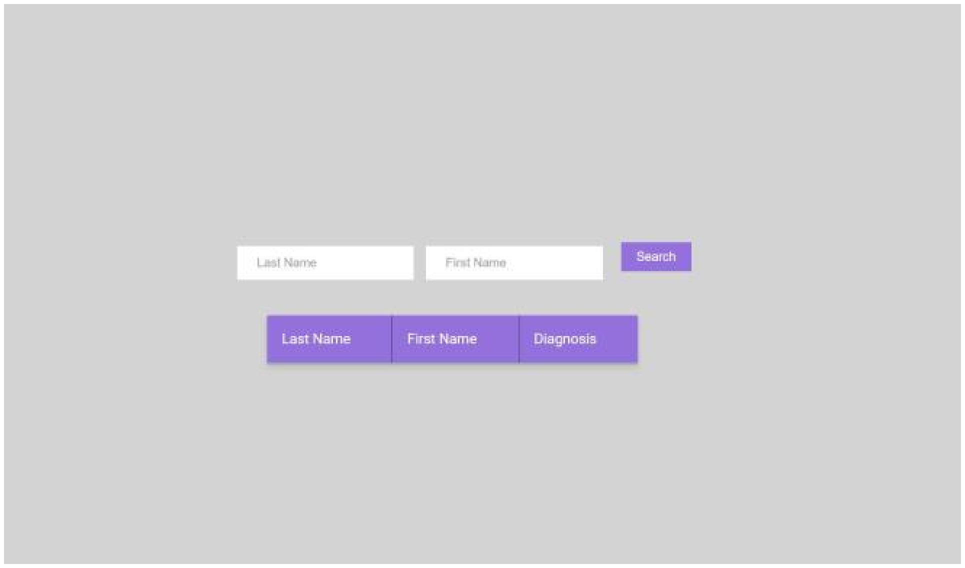
A MySQL database system must be installed, specifications will be provided for schemas, tables, and constraints. Python must also be supported, with the python-hl7 package installed. A Flask installation will also be needed for future functionality.

6.2 External System Dependencies

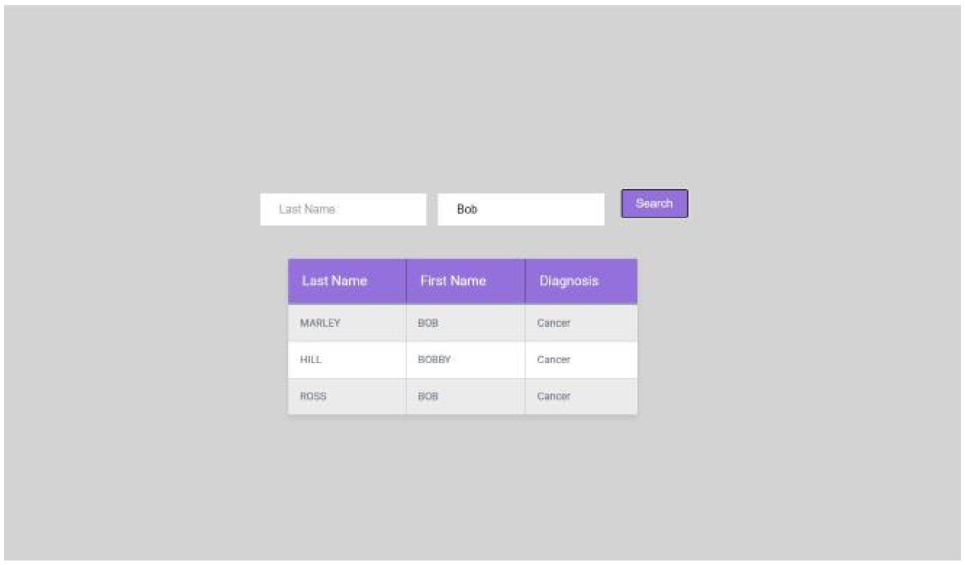
Our system relies on the institution using it to adhere to proper HL7 2.x messaging standards.

7 User Interface

User is presented with a login screen when first opening the app.

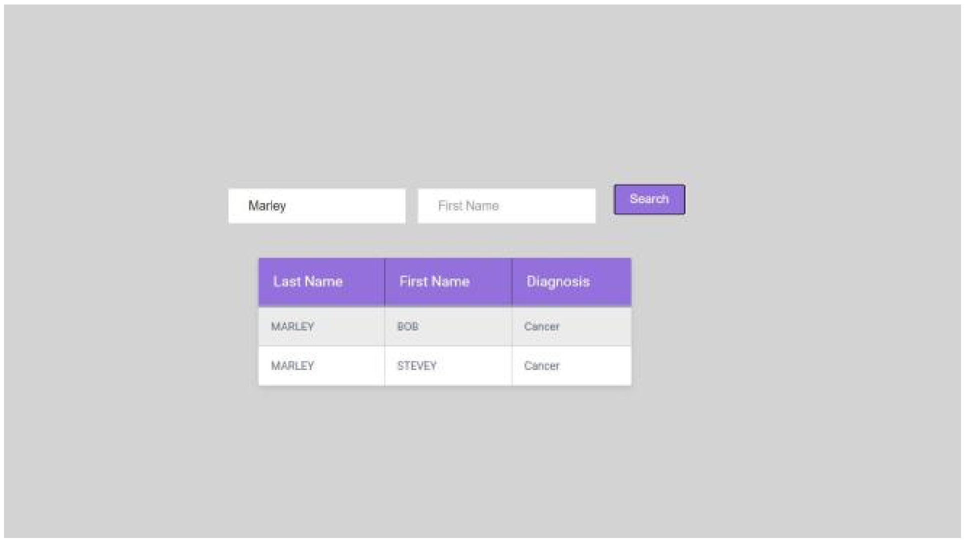


After logging in, user searches for the patient's last name and/or first name and will be presented with patients Diagnosis.



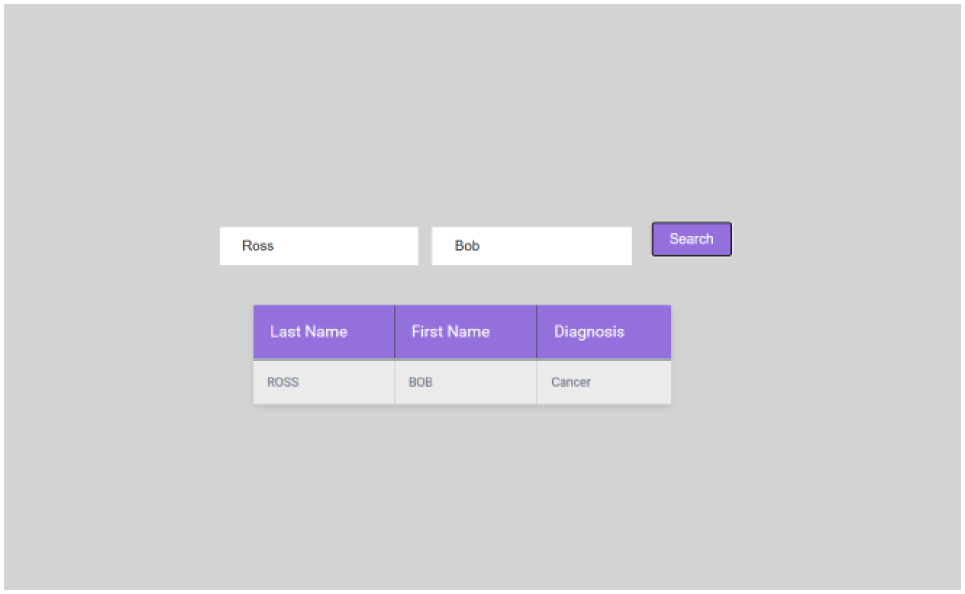
Last Name	First Name	Diagnosis
MARLEY	BOB	Cancer
HILL	BOBBY	Cancer
ROSS	BOB	Cancer

Search by last name.



Last Name	First Name	Diagnosis
MARLEY	BOB	Cancer
MARLEY	STEVEY	Cancer

Search by first and last name



Last Name	First Name	Diagnosis
ROSS	BOB	Cancer