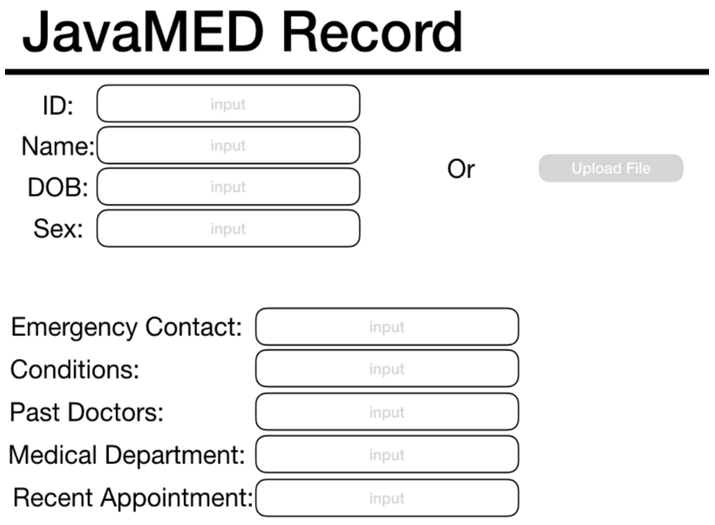
- **Title**: JavaMED Record
- **Problem:** Most interfaces for recording patient medical records are complicated, clunky, and slow in finding a file. Doctors and nurses spend a lot of their time just finding a file in the system and most programs that are available to do this are expensive and managed by a company. In addition, most of the interfaces for these databases are visually unappealing and complicated to learn, making medical staff spend a lot of time learning the tool. To solve this, a novel, open-source, user-friendly software is needed to aid in medical records storage, organization, and extraction

- **Primary stakeholder**: Primary users will be medical staff like doctor, nurses and secretaries. They will use this tool to upload and store patient medical records, send them to the proper medical department, and then retrieve them per request.

- **Graphical User interface**:



- **Data**: As an input, text files will be given that contain the following fields or filled out I the application:
    - Name
    - Date of Birth
    - Sex
    - Emergency Contact Number
    - Conditions
    - Past Doctors
    - Most Recent Appointment
    - Medical Department
    - ID Number

Data Structures: To store these, a **hash table** would be use for each department. However, every medical department will be stored as a **2-3 leaf.** The leaf will contain hash tables as their values and the key will be the string objects (name of department).

Operations: First operation would be to input information into a patient class. This could be done through the interface or by just uploading a text file with all the fields per line. Other operations would be those that come with **2-3 Trees** like, **lookup, insert, delete, print,** and anything else in BST interface. Lastly, hash tables will have a load factor threshold of 0.75 ad starting size of 11. If filled, size will jump to next prime number. Operations for table will be **insert, delete, get, print,** and others found in the abstract data type for hash tables. Patient class will have a specific hash code that will be based on the patient's id number.