# Project Documentation

## Objective

To provide a basic program that allows the user to create, modify, and access patient records. Records includes prescriptions that the patient is currently taking. If the patient is taking more than one prescription, the program will test for any negative major interactions between them and output a warning if necessary. The program will also provide a standalone function to access the drug interaction data by itself for quick reference.

## Purpose

To streamline healthcare delivery by providing a program allowing access to patient records for healthcare providers. Additionally, the program contains functions that ultimately serve to improve patient safety.

## Intended Users

Intended usage is for healthcare providers in a pharmacy setting such as pharmacists, pharmacy technicians, and clerks whom require access to patient and drug information.

## Business Function

Healthcare providers need the ability to store, modify, and search patient records in order to provide services. This program provides usage of these patient records adding, editing, deleting patient records. Prescriptions for a specific patient record can also be added or deleted. Other features such as patient billing information can also be implemented in the future.

## Technical Function

A core part of this program involves patient safety. In every patient record, there is a list of prescriptions that the patient is currently taking. Some prescriptions can cause interactions when taken together and cause negative side effects. These interactions are often categorized by major, moderate, or minor interactions. This program considers any major interactions, which are usually clinically significant, between prescriptions. Thus, if a major interaction is detected, the program will output a warning to the healthcare provider when accessing a record. This checks the validity of multiple prescriptions, helps prevent any mistakes in patient management, and enhances patient safety.

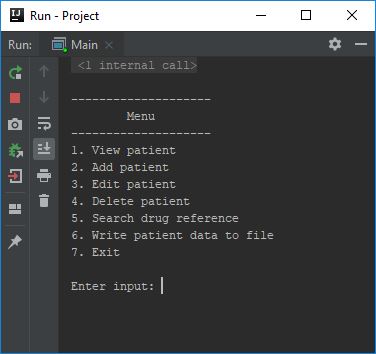
## Miscellaneous Information

Business Type: Pharmacies

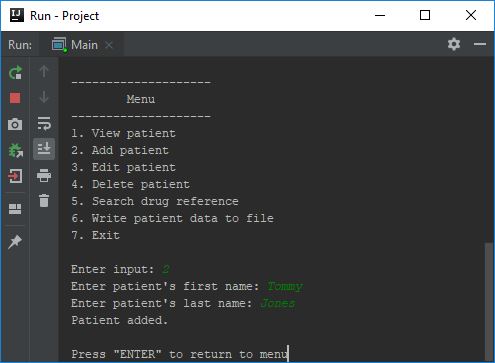
Imaginary Software Company Name: PatientHealth Solutions

## Screenshots

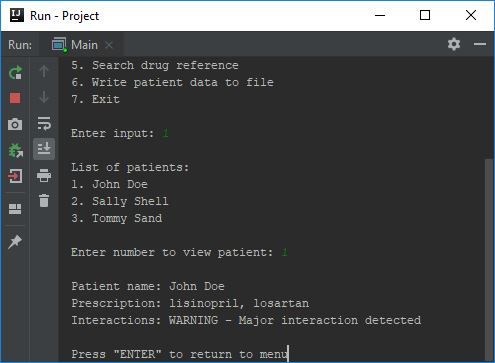
Menu output



Example of adding patient

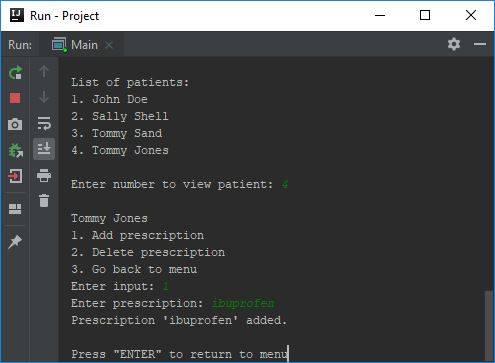


Example of viewing a patient’s record

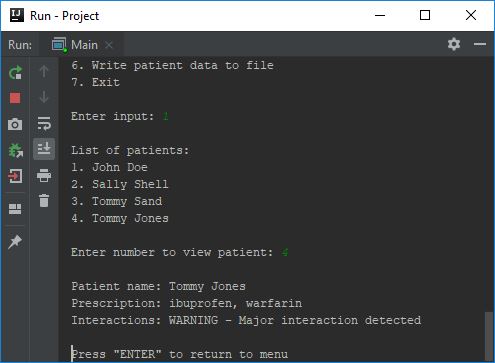


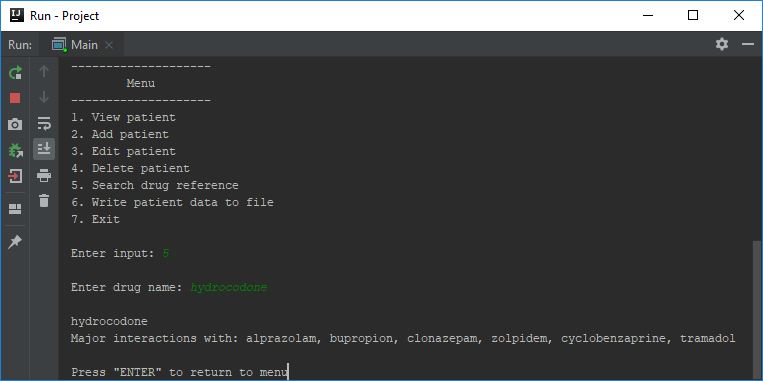
Example of adding a prescription to a patient.

Note: If the user enters a prescription that the program does not recognize, it will be rejected.

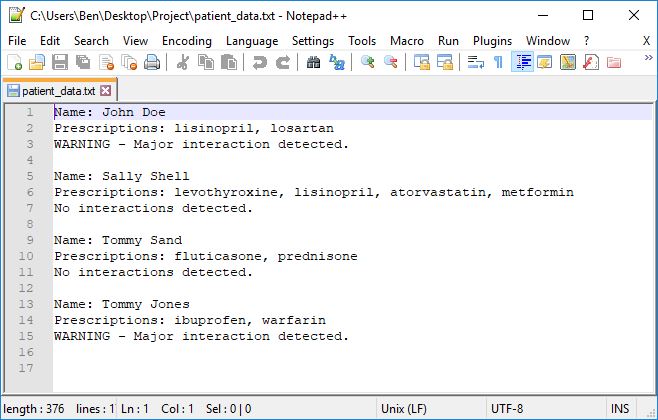


Example of adding prescriptions that counteract with each other



Example of looking up a drug

Example of data output to file



# Pseudocode

**Program start, logic flow:**

Read in data file containing the top 50 drugs and their major interactions with each other.

Load preloaded patient data to populate the program at first.

Output menu with options for the user

List of options include:

1. View patient

2. Add patient

3. Edit patient

4. Delete patient

5. Drug interaction reference

6. Write patient data to file

7. Exit program

Get numerical user input within range of options (Use switch/case)

If 1

Output list of patients with assigned number

Get user input for specific patient

Display patient information (Name, prescriptions, interactions)

Return to main menu

If 2

Get user input for patient’s first name and store in variable

Get user input for patient’s last name and store in variable

Create a new patient using values provided by user

Return to main menu

If 3

Output list of patients with assigned number

Get user input for specific patient

Output a submenu

1. Add prescription for patient

2. Delete prescription for patient

3. Go back to main menu

Get user input for submenu

If 1

Get user input for prescription

Check prescription name is valid

If valid, add prescription to patient’s prescription array

Check prescription interactions\*

Return to main menu

If 2

Display list of patient’s prescriptions

Get user input for corresponding prescription to delete

Check prescription interactions\*

If valid input, delete the prescription from the array

Return to main menu

If 3

Return to main menu

If 4

Output list of patients with assigned number

Get user input for specific patient

Delete the patient from the patient array

Return to main menu

If 5

Output prompt to ask user which drug to search for

Get user input (drug name)

Check if user input is valid, located in the drugs data list

If valid, output drug name and interactions

If not valid, output error

Return to main menu

If 6

For all patients in array list

Write name to file

Write prescription list to file

Write interaction warning to file

If 7

Exit program

**Examples of classes to use:**

Drug class

name of drug

array of drugs that cause major interactions (type String)

Patient class

first name, last name

array of prescriptions (type Drug)

bool warning (to signal a major interaction between prescriptions)

**Algorithm to check for major interactions between prescriptions:**

To check for drug interaction for a patient:

Loop patient’s prescription list (x)

Loop each drug in interaction list from the prescription above (y)

Loop patient’s prescription list again (z)

If y is found in z, (patient’s prescription’s drug matches a different prescription in the patient’s prescription list)

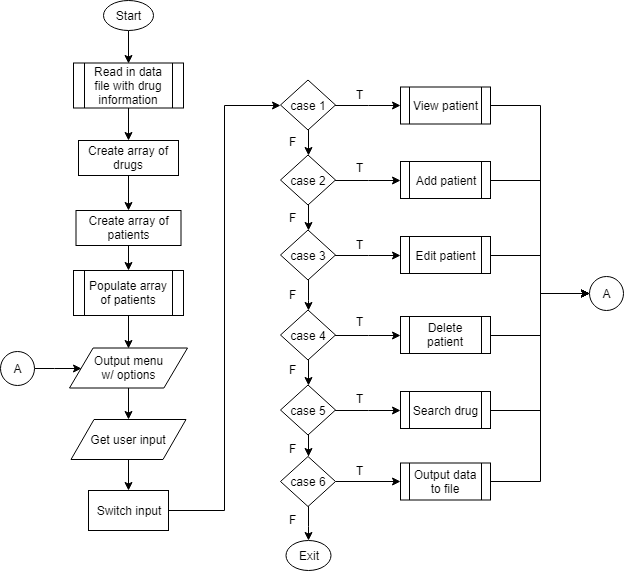
Turn warning on

Break, discontinue loop

Otherwise, warning will stay false

# Flowchart

Program Start / Menu Options



Patient options

