Design Document

CMPUT 291: Mini Project 1

GROUP c291g19: Kelly Chin (1391757), Calvin Ho (1436255), Jessica Huynh (1394832)

Table of Contents

[General Overview 2](#_Toc465342491)

[User Guide 2](#_Toc465342492)

[Initial welcome screen: 2](#_Toc465342493)

[Login 2](#_Toc465342494)

[Detailed Design 3](#_Toc465342495)

[Testing Strategy 3](#_Toc465342496)

[General Strategy 3](#_Toc465342497)

[Test Case Coverage 3](#_Toc465342498)

[Bugs 3](#_Toc465342499)

[Group Work Break-Down Strategy 4](#_Toc465342500)

[Kelly Chin 4](#_Toc465342501)

[Calvin Ho 4](#_Toc465342502)

[Jessica Huynh 4](#_Toc465342503)

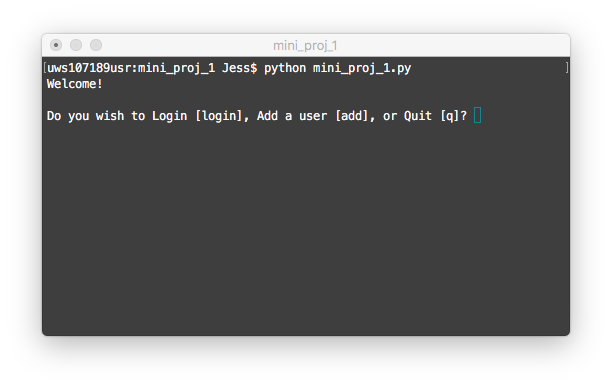
# General Overview

This program implements a hospital database that allows users (doctors, nurses, and administration) to login and execute a number of tasks respective to their role in the hospital. The program also allows a user to be added to the database.

## User Guide

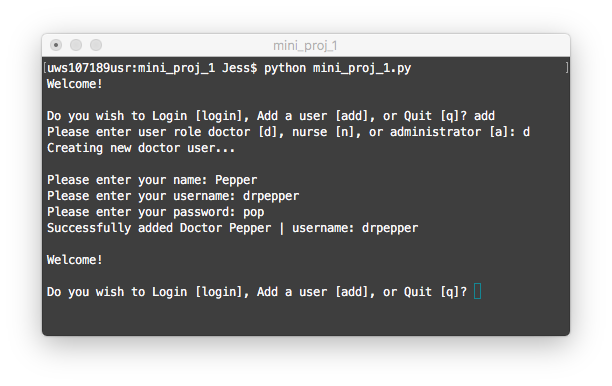
### Initial welcome screen

Prompts the user to select whether they want to login as an existing user, add a user to the database or quit the program.



### Adding a new user

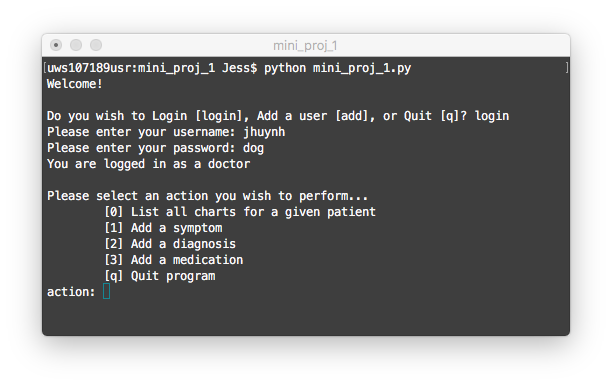
When selecting to add a new user, it will prompt the user to choose the user role they wish to add. It will then prompt the user to input the necessary information (i.e. name, username, and password). Note: if you open the database in a visual manager, the password for each staff will be encrypted.

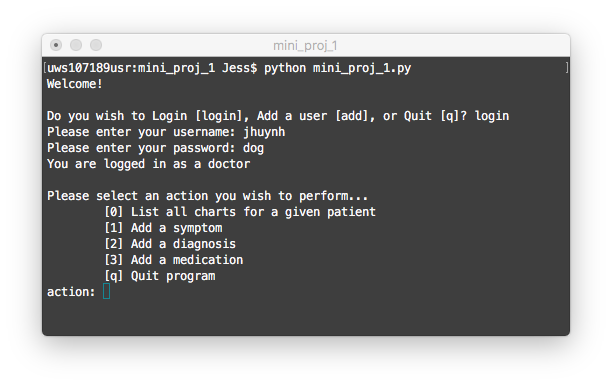


### 

### Login as an existing user

### 





# Detailed Design

## Mini\_proj\_1

### main

first function to be called when the program is started. It calls for other functions such as; ‘promptForInitialAction’ to ask for what action (login, add, quit) to execute, then depending on the user’s response, it will either exit the program if user enter quit command. Calls for ‘promptForLoginInfo’ to ask the user for username and password when user enters login command. When user enters their username and password, the function ‘verifyLoginInfo’ is called to check if the inputted information is accurate and exists. If the information is valid, it calls the ‘userController’ function to display available actions for that user and takes in a command for the action that the user prompted for. If the username and password information is invalid, the user is notified, and prompted to re-enter their information. If the user inputs the add command, the ‘addUsers’ function is called to ask for the new user role, name, username, and password, and then that user is added into the database. Once the new user has been added successfully, this ‘main’ function is called again.

# UserController

Depending on the user’s role. Displays possible actions, and reads the action command the user enters to perform that task. Once the task is complete, it prompts for action command again.

# Testing Strategy

## General Strategy

Our approach to the test the functionally of our code was done in two ways. For the SQL queries that were used to accomplish the tasks for each user (doctor, nurse, and administrator), they had been tested as if they were .sql files. In other words, these queries were tested without integration to a host programming language (i.e. Python). Once the queries passed the tests, they were merged into the python host source files. It was then further tested with the interface to ensure that the correct data was displayed.

## Test Case Coverage

## Bugs

# Group Work Break-Down Strategy

This mini project was coordinated and tracked using Github. Git issues were created, assigned, and closed as a to-do list as each member tracked their tasks. The versioning of different code was addressed using Git branches.

## Kelly Chin

Main responsibilities were doctor tasks.

* Initial login interface
  + Base implementation asking to log in or add user
    - Time spent: 1 hour
* Doctor SQL Queries
  + Task 1: List all charts for a given patient
    - Time spent: 3 hours
  + Task 2: Add a symptom
    - Time spent: 20 minutes
  + Task 3: Add a diagnosis
    - Time spent: 20 minutes
  + Task 4: Add a medication
    - Time spent:

## Calvin Ho

Main responsibilities were administration tasks.

* Administrator SQL Queries
  + Task 1: Create report for all doctor prescriptions
    - Time spent: 1 hour
  + Task 2: List all prescriptions for specific drug
    - Time spent: 45 minutes
  + Task 3: List all possible medications for a specific drug
    - Time spent: 1 hour
  + Task 4: List all diagnoses made prior to prescribing specific drug
    - Time spent: 45 minutes

## Jessica Huynh

Main responsibilities were python interface and nurse tasks.

* Adding a user
  + SQL and interface code
    - Time spent: 10 minutes
* Login as an existing user
  + Error checking & Password Encryption
    - Time spent: 3 minutes
  + Prompting Actions that respective user (D, N, A) can do
    - Time spent: 2 hours
* Nurse SQL Queries
  + Task 2: Create a new chart
    - Time spent: 30 minutes
  + Task 3: Close an open chart
    - Time spent: 30 minutes
* Design Documentation
  + Write up
    - Time spent: