Feature List for Medical Database

David Kelly

December 2015

Compulsory Requirements:

- uses many features of the Java Swing library, including, but not limited to:
  - JTabbedPane
  - JSplitPane
  - JFrame, JPanel, and custom panels using overriden methods
  - JComboBox, JFileChooser
  - JOptionPanes, JButtons, JLabels (many using in-line html) etc
  - action listeners, different layouts (including GridBagLayout, BoxLayout, BorderLayout, CardLayout)
  - Linear gradients, transparency, different border styles
  - JMenu
  - the GUI has functioning cut, copy and paste
- has an encrypted login system
  - the password is not stored in plaintext within the program, but rather as a hashstring generated from an MD5 checksum.
  - for the sake of those looking at this project, the username is **root** and the password **medical**
- Register a new patient
  - it is possible to register a new patient on the database using the "Add Patient" button. This patient will be checked for errors before adding to the ArrayList and writing to file.
  - it is possible to add/remove a patient profile photo and a collection of medical images.
  - it is possible to add/remove a clickable uri to the patient profile.
- Edit a patient
  - special editing panel for existing patients. It is not possible to edit the main screen, thereby protecting the user from making accidental changes. The edited field is also checked for errors before saving.
- Delete a patient

  - it is possible to delete a patient in the database. The user is asked for confirmation before performing this action.

- Search for a patient
  - the search function uses a regular expression to search through the entire database, so it is not limited to searching just by field. Clicking the search button with an empty search field returns the entire database.

- Saving / Exporting
  - the database can be exported to a csv file. This is done automatically when a patient is edited. It is also possible to request this from the main menu.

- Importing
  - it is possible to import a *correctly formatted* database file into the current database. This can optionally be saved to the main database file.

- Javadoc
  - there is a complete javadoc section in the source folder.

- Exceptions
  - try / catch statements are used throughout. Java.logger writes exceptions to the console.

- Links
  - allows any valid uri to be copy/pasted into the database, rather than relying only on the api of one particular webpage.

Advanced Features

- Advanced Reading
  - this project makes use of a number of libraries not found in the lecture notes:
    * Maven: this project uses maven (via the eclipse plugin) for dependency management and jar building.
    * JDatePicker: an open-source library for swing which produces a clickable calendar.
    * Java.security.MessageDigest: for the MD5 password check
    * Java.net: for the clickable uri
    * opencsv: an open-source csv utility library
    * Java.util.HashSet: to ensure that id numbers are not duplicated

    * Junit: this project has a test package for the patient handler methods. All tests are run upon starting the maven package build.
    * java.lang.reflect: this project uses reflection to ensure that the number of setter methods is the same as the length of lines of the database (thereby seeing if either the file or the source code as been manually altered). If it is not, the system exits, protecting any data.

- Optional Features
  - all optional features have also been implemented

The video for the project can be found at : (https://youtu.be/9FzmrSxyoyE) The github source can be found at: (https://github.com/kellino/medicaldb) The jar can be found in the **target** folder in the zip.

This document was prepared using *pandoc-flavoured markdown* and *pdflatex*