Final Presentation

By Darren Huynh
5/18/2016

I am

- 22 Years Old
- Psychology Major
    - (Trying) Going to Graduate School for Guidance Counseling and Social Work
- Video game fan
- Build Computers for friends, experienced in troubleshooting

Computer Science Minor

Reasons why I chose CSCI as a minor:

- Genuinely interested in Computer Programming
- I had taken enough CSCI courses aside from 232 and 233
- More fluent/familiar with higher order programming languages like Ruby, Python, Java than C++, Assembly
- Interested in Content Management/Databases
- Would like to do Front/Back end web development

Overview of Project

My project is an Inventory Program that allows users to:

- Create Entries
- Edit Entries
- View Entries
- Generate a Summary Page

In a user friendly interface. The information is stored in an SQL table, which interacts with Python as its container/integrator.

There is also an "Admin" function, which allows "admin level" usage, which includes deleting entries.

Recap of Goals

Python: Functions to Create Data, Store, Retrieve with exception catches (non-repeating data entries) working properly. Function to grab a computer object and modify it will be included

GUI: Widgets developed, menus, resizing windows and widgets

SQlite: Hierarchical table structure, all data normalized, keys are properly assigned. Many to one relationship network etc.

Goals Met

- Python - Data manipulation functions implemented, error handling is mostly complete, admin login/password to prevent arbitrary data modification.
  - Changelog integrated, records activity, activity type, and time. Changelog is created if it doesn’t already exist in directory.
- TKInter - Button widgets are fully functional, GUI is easy to look at and use, things are placed appropriately. Summary page is dynamically created.
- SQLite - Database/table is created and/or connected to. Manipulation of the table is fully implemented. This (Computers) table is created if it is missing from the directory as well.

TkInter Usage

- GUI application developed using TkInter widgets
- Linked Frame “pages”. Buttons show different frames/pages that are preloaded/reloaded dynamically.
- Individual “pages” have separate frames that allow easier manipulation of widget placement.
- Entry/Label widgets are clear, buttons to interact are in proximity.
- Buttons execute functions that interact with SQLite to perform database operations.
- Message/Dialog boxes pop up to either verify completion, announce errors, and generally communicate information to user.

SQLite Usage

- Database with a table to store information about computers.
- SQL commands take user entry via TKInter buttons, and executes table functions, such as Insert (Create Entry), Update (Edit Entry), Select (View Entry), and Delete (Admin Delete)
- Database/Table is created if it does not exist in the active directory.

Python Usage

- Program written completely in Python(2.7).
- Integration of modules that implement/supplement TKInter and/or SQLite
  - Tkmessagebox, timesys, date etc
- Functions that are called upon button press, and execution of SQL statements are written in Python.
- Overall, Python is the mastermind of the program.

Project Demo

Difficulties Expected

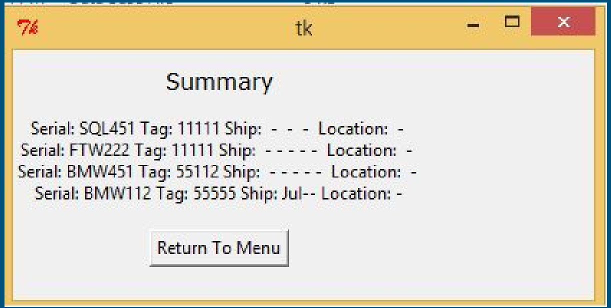
- Ensuring data normalization
- Exception handling
- Functions working properly

Difficulties Faced

- Using Grid
  - Learning Grid was a bit tricky, but I got through it
- Getting column data from SQLite table
  - The way that fetchall() works is weird, as it takes an entire row, and splits it into a list
- Catching weird exceptions
  - At first the view entry page only displayed the first computer that matched the query
- Summary page used to be loaded statically.

Pictures Old Summary

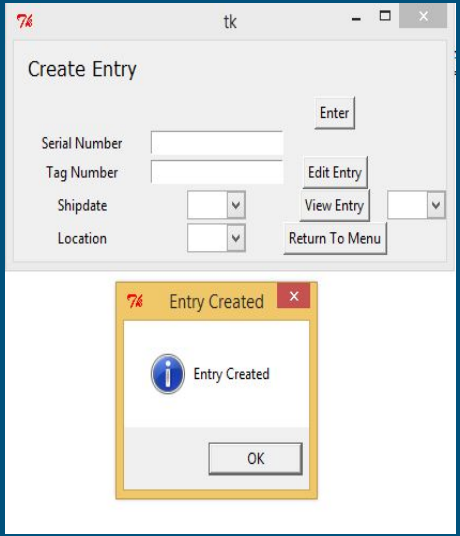
Before Adding Computer



Summary

```
Serial: SQL451 Tag: 11111 Ship: - - -  Location: -
Serial: FTW222 Tag: 11111 Ship: - - - - -  Location: -
Serial: BMW451 Tag: 55112 Ship: - - - - -  Location: -
Serial: BMW112 Tag: 55555 Ship: Jul-- Location: -
```

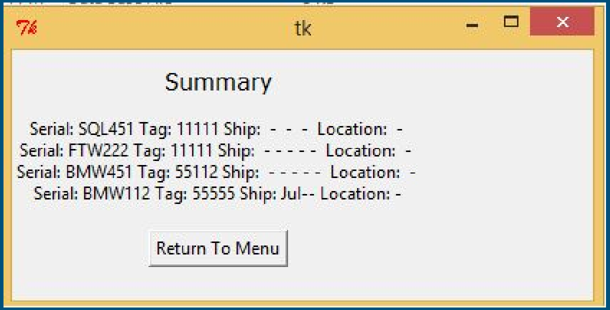
Created Entry



Entry Created

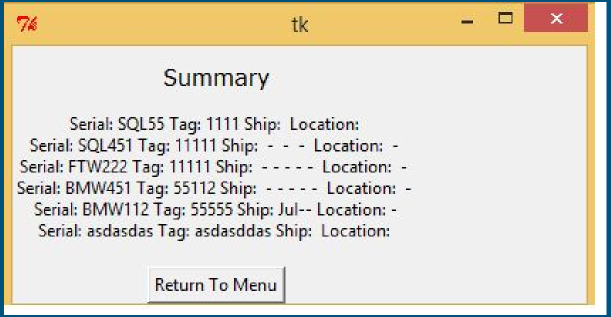
Before Reload

After Reload



```
Summary

Serial: SQL451 Tag: 11111 Ship:  - - -  Location:  -
Serial: FTW222 Tag: 11111 Ship:  - - - - -  Location:  -
Serial: BMW451 Tag: 55112 Ship:  - - - - -  Location:  -
Serial: BMW112 Tag: 55555 Ship: Jul-- Location: -
```



```
Summary

Serial: SQL55 Tag: 1111 Ship:  Location:
Serial: SQL451 Tag: 11111 Ship:  - - -  Location:  -
Serial: FTW222 Tag: 11111 Ship:  - - - - -  Location:  -
Serial: BMW451 Tag: 55112 Ship:  - - - - -  Location:  -
Serial: BMW112 Tag: 55555 Ship: Jul-- Location: -
Serial: asdasdas Tag: asdasddas Ship:  Location:
```

Lessons Learned

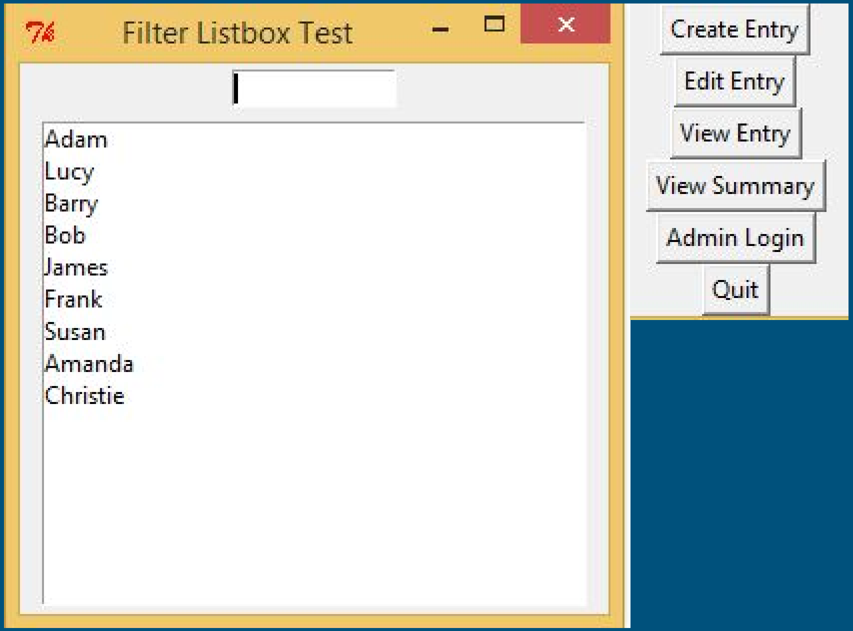
- Grid is superior to Pack for more interactive programs
  - Pack is useful for simple GUI designs, such as single frame, non-interactive windows.
  - Grid lets you place everything where you want, and decide sizes of columns etc.
- Preloading frames is insufficient for updated “pages/frames”
  - Preloading frames in a container object will not update dynamically; using TopLevel() allows for easier window creation that displays information that is changed within a program session. Toplevel is also really useful for Tkmessagebox
- Error Handling is easier to write as you write functions initially
  - Didn’t have to backtrack to catch errors as I wrote exceptions and checks earlier

Possible Expansions

- Conversion of User input into Object Creation
    - Enables object manipulation, class functions, inheritance
    - Instead of grabbing column data, we can grab objects and extract information
        - Allows for “Ticket” status (Using Checkbox for Error - Boolean Column?)
- More advanced GUI management
    - Sizing of windows/frames is not great, implementation of scrollbars for summary page
- More options for summary page generation
    - Different parameters to filter/order by increases utility IE: Show computers on X floor, or shipdates exceeding certain date etc.

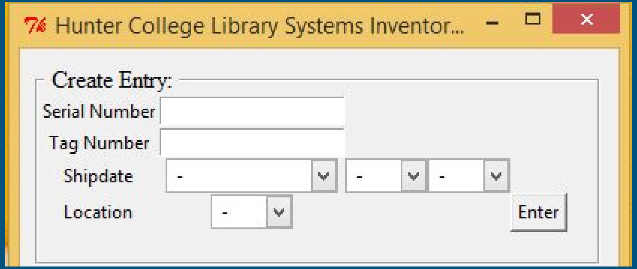
Different Vision/Expansion

- Instead of separate pages, reprogram the whole thing as one big window, with items in a listbox that you can select, and then perform a certain action on
- Will still have some Toplevel Prompts for interaction.



Filter Listbox Test

Adam
Lucy
Barry
Bob
James
Frank
Susan
Amanda
Christie



Hunter College Library Systems Inventor...

Create Entry:

Serial Number

Tag Number

Shipdate - - -

Location -

Random Thoughts

- Coding buttons/labels for each “page” was incredibly repetitive and is ugly to look at
  - A lot of code is not “DRY”. It bothers me a lot.
- Python is somewhat limited in terms of functionality with/without modules
  - I had to find modules that allowed for a function that I wanted, IE messagebox required to make “proper” dialog boxes in my opinion.

In Conclusion

- Learned a lot about GUI programming (In Python).
- Database management is not exactly that fun/clean, but is powerful.