In-Kind Database for Non-Profit Organization

Author:
Kyle Williamson – BS Student, Computer Science

Advisor:
Dr. Alexander Dekhtyar – Associate Professor, Computer Science

California Polytechnic State University, San Luis Obispo
Department of Computer Science
Table of Contents

Introduction ................................................................................................................................................. 3
Motivation .................................................................................................................................................. 3
Background ............................................................................................................................................... 3
Context ..................................................................................................................................................... 3
Problem Statement ................................................................................................................................... 4
Description ............................................................................................................................................... 5
  Data requirements ................................................................................................................................... 5
  User-Interface ....................................................................................................................................... 6
  System Requirements ........................................................................................................................... 7
  Other Requirements ............................................................................................................................ 8
Evaluation .................................................................................................................................................. 9
Design ....................................................................................................................................................... 9
Use Cases ................................................................................................................................................. 10
  Create a new In-Kind record ............................................................................................................. 10
  Reviewing older reports ..................................................................................................................... 13
  Reporting ............................................................................................................................................. 13
  Creating a Roster ................................................................................................................................. 16
Testing ...................................................................................................................................................... 16
Conclusions .............................................................................................................................................. 17
Reference .................................................................................................................................................. 18
Introduction

Motivation

The only reason non-profit organizations are able to exist is through the support of altruistic members of a community. Without proper record keeping and evaluation of donations no company can last for long. Currently records for many organizations are hand-written and stockpiled. The implementation of a database for data storage of donations provides many benefits.

Background

This project became possible when the non-profit organization ALPHA approached the Computer Science department at Cal Poly to ascertain if they could create a database for In-Kind donations. The non-profit provides practical assistance during a pregnancy and through to a child’s first year, as well as education on pregnancy and early-parenting (slo alpha). They provide services throughout San Luis Obispo County but their focus is the city of San Luis Obispo.

Context

In-Kind donations are contributions of time, service, or goods to an organization (SIL International). They are used alongside cash donations and generally accepted if the purpose of the good suits the organization. Some examples include building space for a board meeting, any form of volunteering, and baby supplies. The sheer variability of possibilities for In-Kind adds to the complexity of storing information on goods versus monetary compensation.
Problem Statement

Jen Miller, a manager at ALPHA, proposed the project to the Computer Science Department to digitize their future In-Kind documentation. Volunteers at ALPHA have little computer experience so the project must have a user-friendly interface and simple methods to carry out commands. With an efficient application the organization will feel more comfortable asking prospective donors for goods as well as cash.

The project on behalf of the Computer Science department is creating a fully contained system for creating and reporting on In-Kind gifts. The main tasks for implementation are a simple procedure for recording and retrieving. Side benefits include automatic backup of data and streamlined generation of reports for financial tracking.

The main application will interface with a database created specifically for In-Kind donations. It will also use data from previously created databases already used by the organization for storing information on donors and clients. Volunteers can use the actual Access database for storing information however the front-end of the project will provide conveniences not implemented in Access. The project user-interface must be friendly to those unaccustomed to the charts of data in Microsoft’s database application.
Description

Jen Miller provided some unique requirements along with the requested features provided in the Problem Statement section. The organization is using a legacy Access System created for storing monetary donations of regular donors and clients. To reduce the problem of data duplication errors the project will interface with the existing information networks. The additional risk of relying on an outside program was outweighed by the benefits.

Data requirements

The managed information is displayed in the following UML diagram.

Image: UML design of the In-Kind database
Category relates to the type of In-Kind donation offered. The Category field is a broad area such as Facility and Equipment. Description is more defined and includes volunteer time and Equipment supplies. InKindDonation includes a foreign key to the existing database to retrieve information about the user for reporting instances. The two main uses of the external database are to retrieve only active members for display in the In-Kind database as well as displaying addresses and names for creating reports.

**User-Interface**

In the legacy databases the interface was created inside of Microsoft Access. For this project we wanted to keep the same general layout to ensure a smooth transition for the volunteers that are already familiar with that database. Below is an example of the old database front-end and the new one developed in java.

![Screenshot of ALPHA database main screen](Image: Screenshot of ALPHA database main screen)
Through discussions with ALPHA we determined that there are three main fields required in the In-Kind database. People contains two fields: roster, which display a list of people without clients who have donated goods, and labels, which creates mailing addresses for the respective donors. In-Kind is the main feature of the new application. Edit links to a new page where the user can create a new record or cycle through all existing files. Report to Date simply creates a CSV file and pops a notification message with a directory link to the file. Report Any Date adds another step by linking to a new interface page. Here the user is able to specify a date range, thus restricting the data dumped to the CSV to only those In-Kind donations provided within a specific range. Finally, News Letter creates a list of e-mails of donators in another file and then shows a notification message to the user.

**System Requirements**

For successful installation and use of the program several factors are required. The first condition is installation of Java. The latest known working version as of this report is JRE6u24. Another qualification is an Access database, everything was tested with the 2003 version but it should work with newer versions as long as the data
structures are kept the same. The program relies on the databases for clients and people existing in a location known to the organization and the field structure stays consistent. The final requirement is the latest version of the executable jar AlphaInKindDatabase.

The application exists on a single windows machine which all others access through a network connection. Since a few thousand records will be sent during creation of the interface for display a landline connection will work best, but a wireless system will also function, albeit slower. All of the computers that access the application as well as the host machine are required to run Windows. Supported operating systems are Microsoft XP through Windows 7.

Other Requirements

There are additional requirements to consider besides system. First there is no security, if someone does not have a valid reason for looking through records they should not be allowed to use the computers. Also, because of locking permissions, if the access table is opened the program will not be able to add or fetch any new records.
Evaluation

This project lasted for two quarters. The first 10 weeks consisted entirely of planning and design work. The database design was finalized near the end of the quarter, but the user-interface was an entirely different matter. A hand drawn mock-up was created and approved during the design phase, but throughout the implementation the user-interface was changed many times. By the end of the project the final view was fairly different than the initial approval.

If judging project success by completion of the tasks then this project accomplished its goal. All of the Use Cases for the project were completed and installed and the members at ALPHA seemed grateful for the project. The organization engaged in user testing after the Beta release and they found relatively few bugs. However, since this project is meant for immediate use in the organization, the best way to calculate success is their use of the program in the future.

Design

Architecturally the application runs around a main interface that responds to button clicks. If one of the buttons creates a new screen that view is created and set visible from the main class. After its work is completed the screen is hidden until needed again or the application is closed. The only other class is a custom exception used for catching incorrect data entered in the edit In-Kind donation class. All of the graphical classes were created using visual editor, an eclipse automated GUI building plugin.

The database was created with Microsoft Access and links the tables InKindDonations and Category together. Each category item consists of two parts. The first string, called Category, is used for linking together donations of the same broad generality. Second is the concentration within the field named Description. One item included with the program is Volunteer – Tech, where they plan on counting my time towards their In-Kind donations budget.
To link Java and Access together the program implemented JDBC (Java DataBase Connectivity). This API defines how a client accesses a database with SQL commands. The operation begins by creating a connection with a string defined destination to the database. At that point, using a Statement which functions as a SQL command, is executed and if successful responds with a ReturnStatement which is iterated through to collect all the information. Due to difficulty with the JDBC and complex queries, this program queries the broadest set of data and then performs data manipulation on those sets.

The entire project was designed in small steps along with frequent customer visits. The first deliverable during the implementation quarter was the graphical layout without any of the functionality. After that approval step, the main task became creating the database and integrating the application. That program evolved into the beta release. The last few weeks of the quarter were constructing reporting apparatus. However because of communication issues, uncertainty on ALPHA’s side, and assumptions on the part of the developer there were constant small changes along with each major release.

Use Cases

A use case is the description of steps required to fulfill some useful task. The following four cases show the implementation of the major requirements for a front-end to a database. Included are examples, with personal information blacked out, of adding to the database, retrieving from the database, and generating reports. There are additional smaller cases but they reside in the broad categories. For instance, a case exists for adding new records, where the user can click on a dropdown menu and the program will group categories together based on their use.

Create a new In-Kind record
Pictured here is the page for creating a new record. There are a total of 11 fields,

- **Log No.** is an automatically generated unique number retrieved as the primary key from the database.
- **Name** is a list of all active members alphabetically sorted by last name. It can be switched between regular donors and clients by clicking on the button next to it.
- **User Code** is the unique code of the currently selected name and is updated automatically.
- **Unit(s) and Unit Value** together form the **Total Value** of a single record. This allows ALPHA to allocate multiple equivalent items of a gift into a single record. If the user tries to calculate when both fields do not contain numeric numbers the program displays a warning message. When successful, the **Total Value** field is filled in. In the above image the program is demonstrating the ability to catch rounding errors.
- **Current Total** is calculated by summing the product of all Unit and Unit Value pairs. Total Value is not stored because it is redundant data.
- **Class** is created by joining together Category and Description from the Category table. It arranges all items around a Category and then subgroups the Description string.
- **Date** is automatically generated from Java’s current time. The program will accept any date with the same format but will catch invalid dates during database insertion.
- **Comment** is a required field for the record. Any string under 255 characters is accepted.
- **Make and Model #** is intended for use with only equipment categories, but they did occasionally use the field for other information.
- **Recall** is used for items that are later declared dangerous by their maker. It is used to check quickly if the aid recipient has been notified of the need to return the item.
- The remaining items are used for creating or searching through records.

Once the user has filled in all the fields they need to they can press **Create** to attempt to insert the query. There are four required fields before the program will even attempt insertion. **Total Value, Date, Class, and Comment** need to be filled with their respective data types or the program displays an error message clarifying the issue and waits for the user to respond. If the query is then successful a success message is shown and the application readies the next query.
Reviewing older reports

![Image: Reviewing past data](image-url)

Reviewing past donations inserted into the database is a simple process. Here the main point of interaction for the user are the buttons along the bottom of the screen. In order they are: first record, previous records, the number of the current record, next records, latest records, and create new record. All of the fields are locked and the issue with Unit(s) and Unit Value has been fixed. Similar to inserting items, the records are split up by client donors and regular donors. This image is of the latter. The information is all generated by retrieving information from the In-Kind donation database except Current Total, which is calculated according to the earlier rule.

Reporting

Reports can be generated in two fashions, either through **Report to Date** on the main screen or **Report any Date** which initiates a new screen. In either case, when the report is successfully generated a notification message displays an alert of the creation
location. ALPHA’s requirements include the ability to create a report for Excel. From there they can easily manipulate the data. The initial concept was to use an Excel API to directly convert the data to an .xls file. Although the methods do exist, it was decided to implement CVS (Comma Separated Value) files for the sake of simplicity. Excel is able to open these files and propagate them in its structure, but .cvs files inherently lack formatting information.

The fields on this page are automatically with the current Java date as the **End Date** and one month prior to that as the **Begin Date**. As long as the dates follow the specified format and are possible, a query will be created. Otherwise a warning message showing the problematic field(s) is created.
Testing was conducted with a copy of the databases after signing a non-disclosure agreement. In light of that, all personal information has been blacked out.

The data is split according to Categories based on ALPHA’s specifications. Information is retrieved from querying the In-Kind database, followed by querying the People and Client database with the relevant user codes. The personal information is then added to the .csv file along with the In-Kind data.
Creating a Roster

The last use case is creating a roster of all clients and donors that gave goods to the organization. The descriptions on the left are the current databases categorizations of donors. If the Include text box is blank or Print All is selected the roster will include all fields in the document. Otherwise double-clicking the names will add filters to the data. There are two options for ordering the material, last name and code. Roster Title will name the .csv file everything is saved to. Although there were plans for Print Roster currently it functions exactly like View Roster where it creates the external file.

Testing

There are two main issues of concern that testing covered. The first is data retrieval. This was achieved by providing strict queries similar to larger ones in use. Then the collected data is tested for validity. The second issue is display of data. Although testing covered parts of this a lot of information had to be checked by hand. For instance, near the end of the project it was noticed that in a report all the In-Kind database information was used correctly but information was aligned with the wrong user personal details.
Conclusions

The project began as a database design and ended up focusing more on the user-interface side of programming. It was a sudden change of pace from Cal Poly life to discuss technical details with laypeople. We both had different ideas on the project although in the end we managed to merge everything together. I also realized the benefits of Agile programming when working with a customer since new ideas or forgotten use cases often pop up during a demonstration.

From the project perspective the final solution was completed although it will require continued maintenance. At this point ALPHA has gained the benefits of a digital database for storing In-Kind records. The information is now more secure and easier to work with. Using a Java developed interface instead of hand-writing the ALPHA volunteers will have an easier time recording required information.
Reference

http://www.sloalpha.org/