COMP4 Coursework

Analysis

Introduction

Client Identification

My client is St. Andrew's Street Baptist Church, the administration consists of a team of up to five people of various ages with little experience with computers apart from performing basic tasks such as word processing and using simple spreadsheets. Currently the church has a simple file sharing between a small network of 5 computers, these computers include Apple and Windows computers.

All of the members of the administrative team at St. Andrew's Street Baptist are required to attend various meetings and complete certain tasks, all of which have potential to be confidential. The team provides pastoral care and other supportive services to people all across Cambridgeshire and sometimes further. Being a large city centre church, the membership and the composition of the administrative team is subject to change on a regular basis.

The Church has expressed a desire to use a computer system to organise, centralise and control all of the data involved in operating the Church. With the proposed system, the admin team would like to be able to keep a record of all of their meetings and tasks as well as keep track of the quantities of a few finite resources used in the offices.

Define the Current System

The current system in place is a manual paper based system which involves each member of the administrative team writing down details of meetings, appointments and tasks and recording it in a personal dairy and/or planner. This information includes: a title for that meeting, appointment or task, who else is meant to attend, the location of the meeting or appointment and the date & time at which this meeting will take place.

If a meeting involves more than one person (as most meetings do!) then the team relies on either verbal or email communication of the key pieces of information for that meeting, the person receiving the request for meeting then replies to that email to confirm or deny their attendance to the meeting, if the accept the meeting, they then add a copy of the information to their planner, if not they archive the email and take no further action.

There is also a central list of the resources available to the office staff which is updated on a regular basis with the new amounts of the various resources. All members of the administrative team have access to this list, and there is shared responsibility of who goes to purchase additional resources if the current stock is depleted.

Peter East – 6303 – COMP4 Coursework

Describe the Problems

The current system should work well, in theory, however there have been many occasions when a meeting has been missed because an email failed to send or the organiser forgot to tell them. Often the office has ran out of a particular resource because nobody updated the list or nobody saw that they were running out. Each person has a separate copy of each meeting that they are due to attend which could (and often does) lead to inconsistencies in the information that each person has, also the duplication of the data requires a large amount of physical space. The Church also plans to move it's offices so having a large amount of data increases the risk of data being lost or damaged during the move. Furthermore, the data is not stored in a secure location, it is either left in an office which is not always locked during the day, while the Church is open to the public or it is with it's owner who is liable to drop/forget/lose it which means the data is not secure.

Insert the questionnaire here.

Investigation

The current system

Data sources and Destinations

In the current system, there is a definite flow of data between the different people working in the office, each person is both a source and destination of data. When a person creates a meeting they will write it in their own personal diary, and then they will either email or speak the information about that meeting to whoever they are requesting to attend. The person who is being asked to attend sends information accepting or rejecting the request of attendance back to the person who 'owns' the meeting.

Source	Data	Example Data	Destination
Meeting Owner	Meeting title, meeting location, meeting date & time, people requested to attend the meeting	Coffee with Steve; Ian's Coffee House; Tuesday 13th October 10:15am; with Steve, Joel and Sabrina	Meeting Attendees
Meeting Attendee	Meeting title, Confirm Attendance?	Coffee with Steve, Attending	Meeting Owner
Task Owner	Task title and a brief description of the task	Refile the copier paper; Make sure you use the yellow paper for this week's service sheets.	A scrap of paper pinned to a noticeboard, or a blank page in a personal notebook
Team member	Resource name, resource quantity, resource cost etc	Teabags, 50, 4.30	The resources book
Resources book	Resource name, enough of this resource available?	Teabags, no	Any member of staff

Algorithms

In the current system, there are a few basic algorithms used. The first of which is for checking if a meeting will have all of the requested people in attendance.

```
AllAgreed ← True
Attendees ← [A list of attendees]
FOR index ← 1 TO length(Attendees) DO
      IF(Attendees[index] = False) THEN
            AllAgreed ← False
      END IF
END FOR
IF(AllAgreed = True)THEN
      Meeting goes ahead
ELSE
      Alert meeting owner that not everyone has responded
END IF
```

Another algorithm used has the purpose of checking if someone has completed a task they set themselves, this also ensures that time is not wasted between tasks.

```
TaskComplete ← False
WHILE TaskComplete = False DO
      IF Busy = False THEN
            CompleteTheTask()
      END IF
END WHILE
```

The third, and final algorithm used by the team is to check if there are any resources that need replenishing.

```
ResourceInNeed ← EmptyList()
Resources ← [A list of resources & their quantities]
FOR index ← 1 TO length(Resources) DO
      IF Resources[index][RequiredAmount] < Resources[index][CurrentAmount] THEN
            ResourcesInNeed.append(Resource[index])
      END IF
END FOR

FOR index ← 1 TO length(ResourcesInNeed) DO
      OUTPUT ResoucesInNeed[index]
END FOR
```

Data Flow Diagrams

Key:

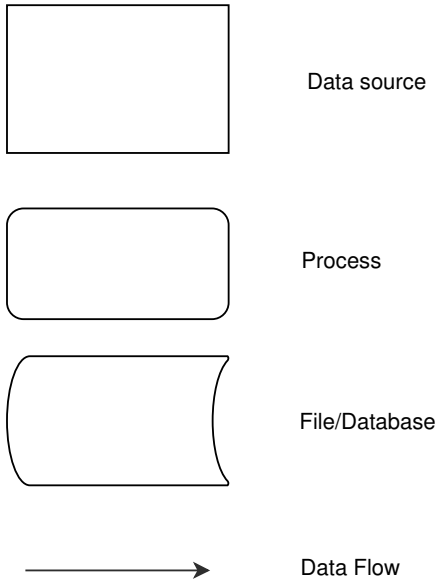
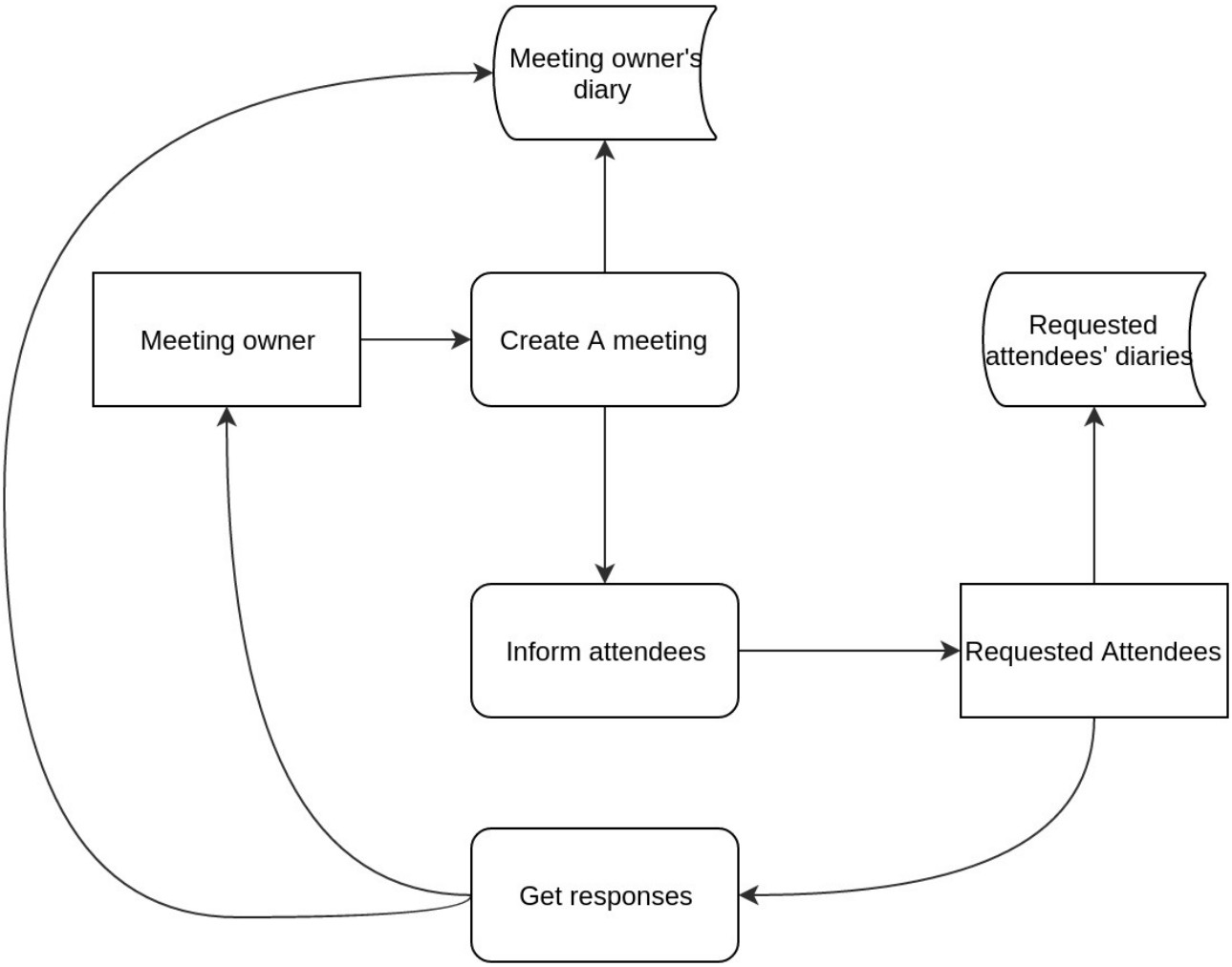


Diagram for the meetings subsystem:



Peter East – 6303 – COMP4 Coursework

Diagram for the tasks subsystem:

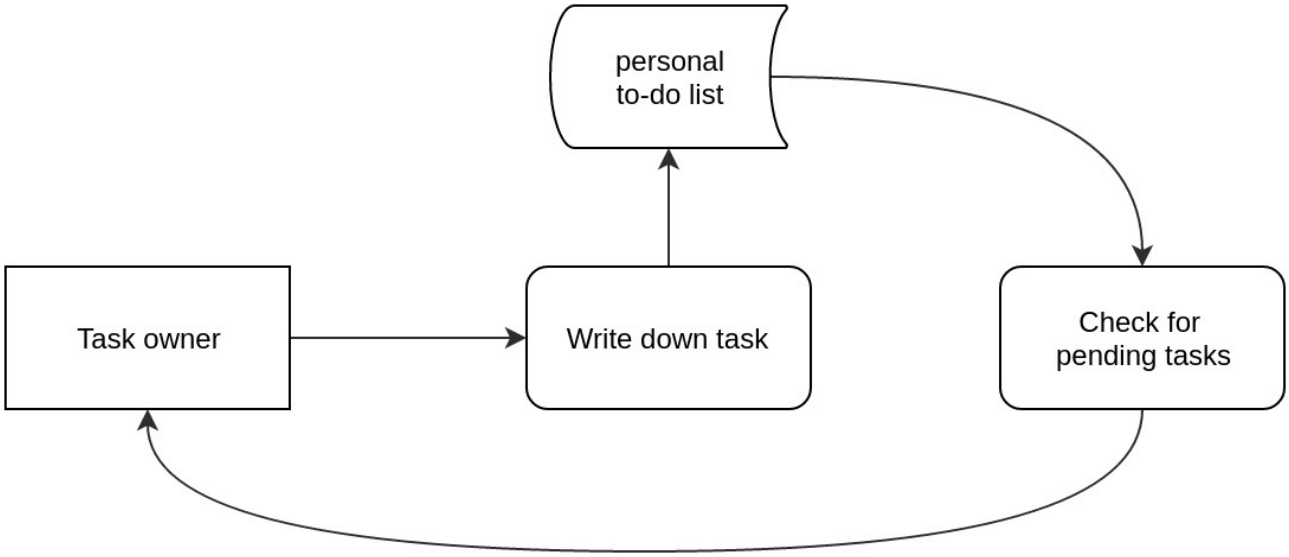
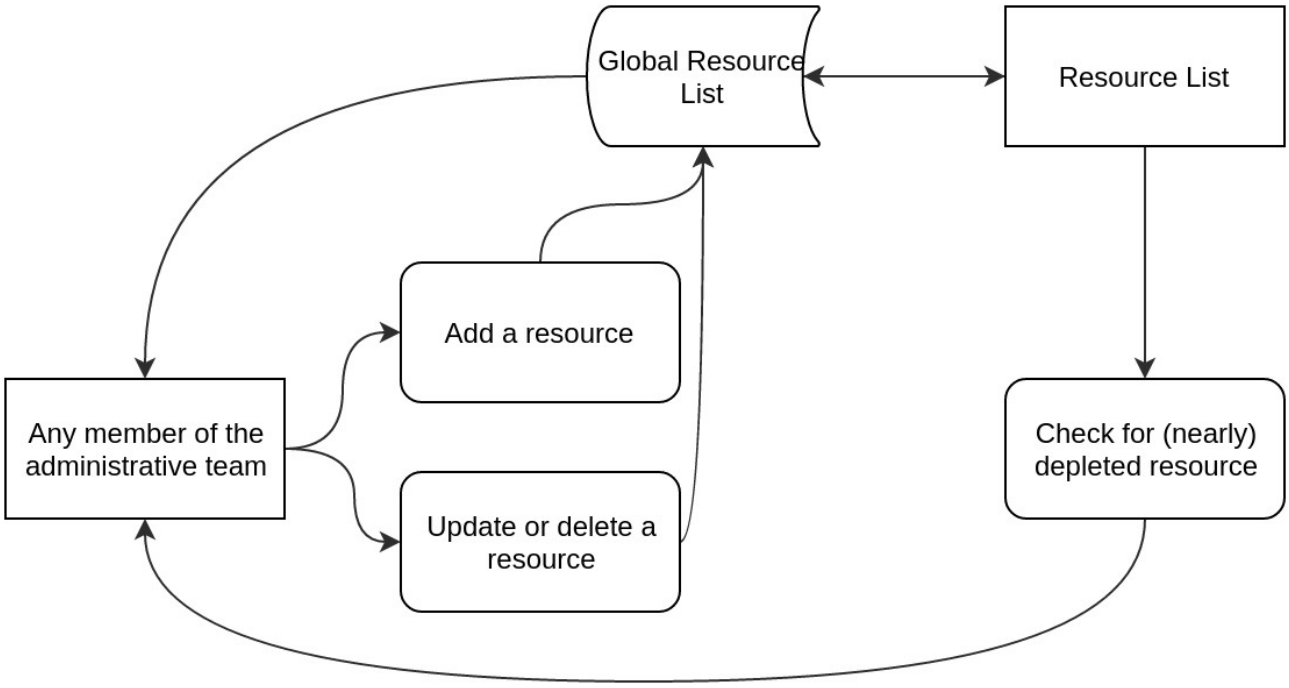


Diagram for the resources subsystem:

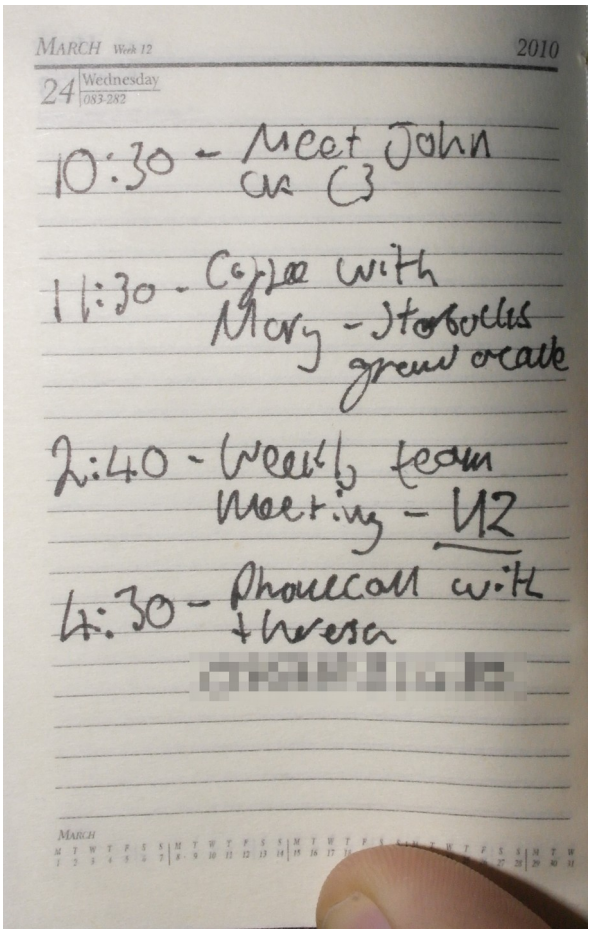


Peter East – 6303 – COMP4 Coursework

Input Forms, Output forms and Report Formats

The current system has three input forms – The team member's personal diaries, the general format used for making notes of tasks and the record book containing the list of resources. The current system has similar output forms, for meetings, often the members of the team give each other photocopied pages from their diaries and for tasks the output form is identical to the input form.

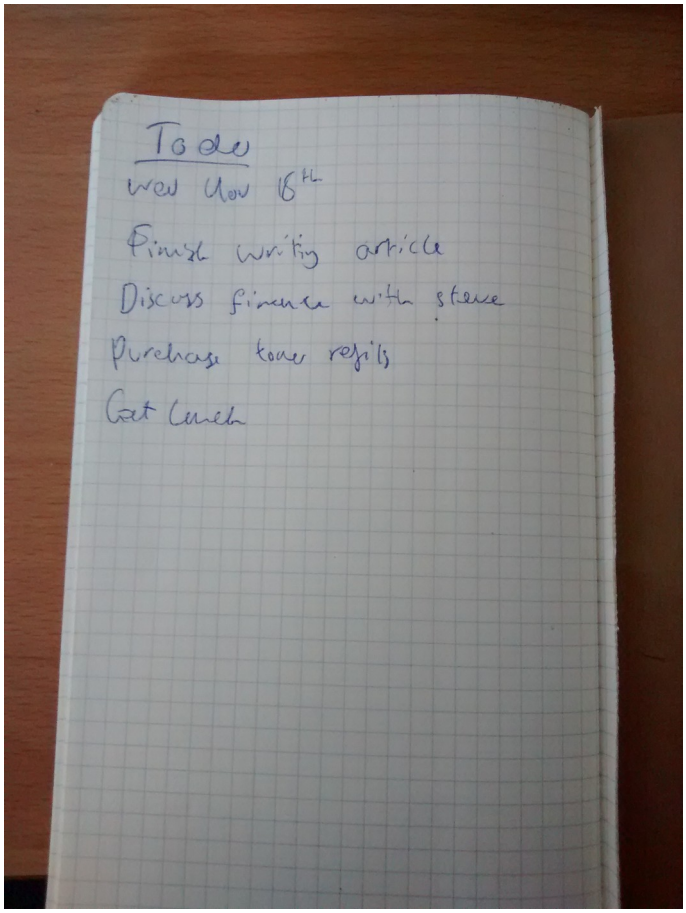
Below is a copy of an old (but still using the same system of notation) diary from one of the supporting staff at the Church, this is both an input and an output form. It contains a list of the day's appointments, with their date & time, a brief subject line and the location of the meeting, as well as any other attendees. Unfortunately



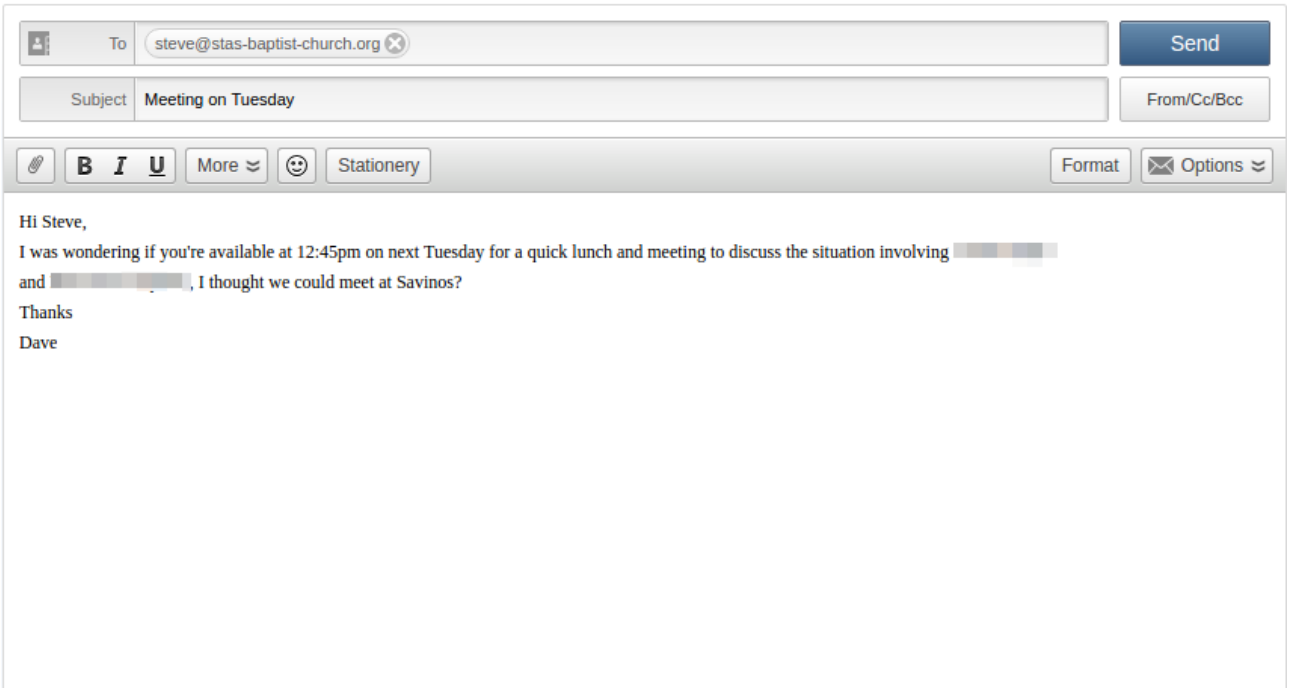
the latest record I was able to obtain was from 2010 as often the contents of the diaries are strictly confidential because the Church is responsible for caring for many vulnerable people – hence some of the information has been omitted.

Peter East – 6303 – COMP4 Coursework

Below is an example of a list of tasks from one of the office staff's notebook, it contains a list of tasks that they need to get done that day. The team also uses Post-It notes, record cards or digital reminders on their phones as alternatives to having a list of to-dos.



Here is an example of an email requesting someone to attend a meeting:



Note that the message is concise and to the point, the staff team deals with hundreds of emails per week and none of them have time to read through a detailed explanation for each meeting they're going to attend.

The proposed System

Data sources and destinations

In the proposed system, the users will input the information for their meetings, tasks and resources into various forms.

Source	Data	Data Type	Destination
Meeting Owner	Meeting Date & Time	String	Database – Meetings
	Meeting Title		
	Brief meeting description		
	Location		
	Requested Attendees – UserID	Integers	
Database - Meetings	Meeting Title	String	Requested Attendee
	Meeting Date & Time		
	Brief description of meeting		
	Location of meeting		
	(Other) requested attendees	Integers	
Requested Attendee	MeetingID	Integer	Database - Meetings
	Confirm Attendance?	Boolean	
Database – Meetings	MeetingID	Integer	Meeting Owner
	Confirm Attendance?	Boolean	
Task owner	Task Title	String	Database – Tasks
	Task Description		
Database – Tasks	Task Title		Task Owner
	Task Description		
Member of staff	Resource Title		Database – Resources
	Resource Cost	Integer	
	Resource Current Quantity	Integer	
	Resource Required Quantity		
Database – Resources	Resource Title	String	Member of Staff

Source	Data	Data Type	Destination
	Resource Cost	Integer	
	Resource Quantity		
	Resource Required?	Boolean	

Data Flow Diagrams

{do these later}

Data Dictionary

Volumetrics

There are 5 members of the staff team which will have between 2 and 5 meetings per day, which means in one month, the system will have up to 750 meetings, each of which will require up to 1815 Bytes, which will total at 1.3MiB per month for the meetings system. However, the number of people who will be involved in meetings will be subject to change on a weekly basis, it could easily double, triple or more which means that this part of the database could use up to 5MiB per month and up to 60MiB per year. The user's table will contain a record of each user who has meetings, which totals at 798 Bytes per person. In a large city centre Church, there are approximately 250 congregates, of which 50 will be involved with meetings, which means the user's database will require at minimum, 39KiB. However this number is subject to change and could easily double in the space of a year, but because of the data protection act, some means of ensuring the data does not expire would mean that as, or shortly after new users are added, the old users are removed so the size of the table will only fluctuate by $\pm$10KiB, so to ensure there's always enough space, this table will be allocated 60KiB. The tasks table, which is effectively a record of each user's to-do list, will contain records for between 5 and 10 users each with up to and estimated 15 items per day. Each item is 5170 Bytes, which means every day, each user will generate up to 76Kib, with up to 10 users totalling at 760Kib, so in a month the database will contain up to 22.8Mib. However, the nature of the tasks means that they will expire after a certain amount of time, which will limit the size of the database. If the expiry period of each task is set to one year, this table will not exceed 273.6MiB. The resources table will contain a record of all of the things the Church regularly buys, including cleaning supplies, food, drinks, cafe supplies. The church has several areas which require resources, some of which require more than others. If in total, the Church, buys 300 consumable products, each record will require 271 Bytes, with 300 items, the table will require 80KiB, however there is potential for new items to be added on a regular basis, so this table could easily grow to 100KiB.

The total database requirements are 334MiB, for a year's worth of data. The program itself, the PyQt Library and Python will require about 130Mib, which means the total estimated size of the system will be 464MiB.

Peter East – 6303 – COMP4 Coursework

Objectives:

General Objectives:

- A simple and clear layout structure for viewing recorded meetings.
- A simple and clear layout for adding new meetings.
- A simple and clear layout for adding and viewing a to-do list of tasks.
- A clear and effective way of monitoring stock levels of various resources.
- A way to edit the user information

Specific Objectives

- Viewing meetings:
  - A clear and consistent structure used for displaying meeting objects.
  - Minimal controls to ensure accessibility and to reduce the necessity for training.
  - Only essential information shown.
- Adding meetings:
  - An input structure that follows a pattern similar to how the meetings are displayed.
  - Easy selection of attendees from a pool of available users.
  - Validation of the user's input.
- Adding and Viewing To-Dos:
  - A clear, prioritized list of tasks
  - The option to mark tasks as “Done”
- Viewing and editing resources:
  - An ordered table of information for all the recorded resources
  - The ability to add additional resources at any time
  - The ability to update the quantity of resources available.
  - A way to quickly view a list of resources that are below the required level.
- Editing user information
  - A way to change the user's password

Core Objectives

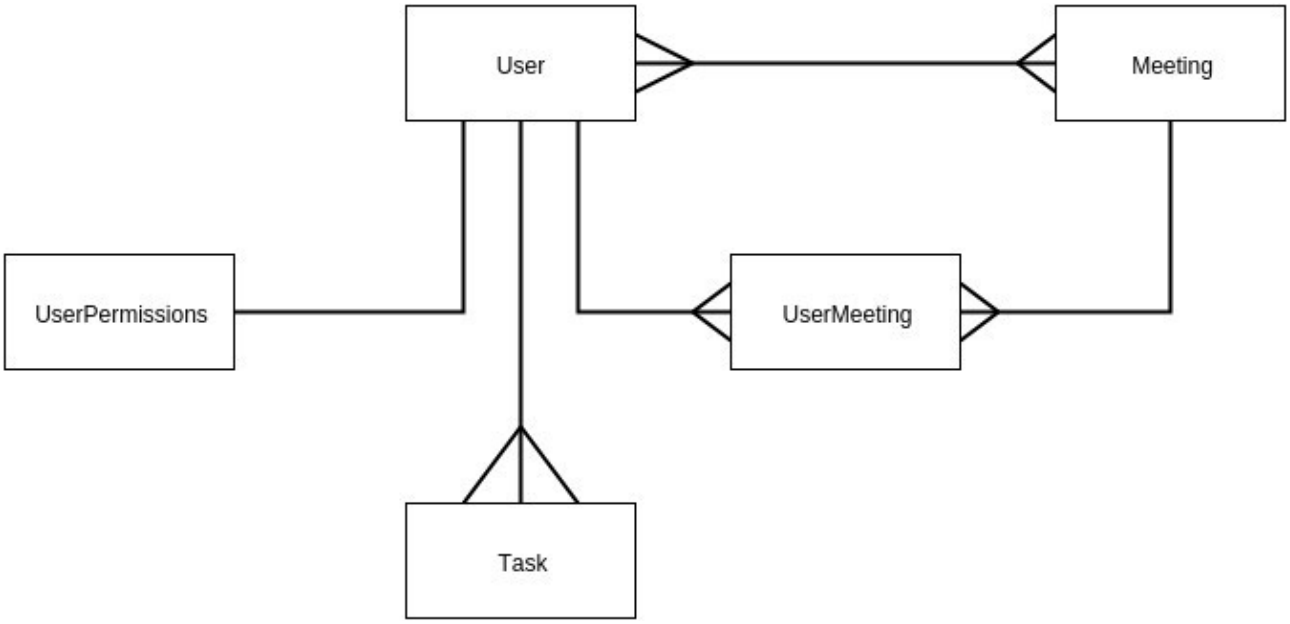
- Meetings viewing/adding

- Tasks viewing/adding
- Monitoring Resources

Other Objectives

- Editing user data

E-R Diagrams & Descriptions



Entity Descriptions:

User(UserID, Username, UserFirstname, UserLastname, UserPasswordHash, Permissions)

Meeting(MeetingID, MeetingOwner, MeetingTitle, MeetingDateTime, MeetingPlace)

MeetingAttendee(UserID, MeetingID, UserMeetingConfirmation)

Task(TaskID, TaskTitle, TaskDescription, TaskOwner, TaskExpiry, Priority)

Resource(ResourceID, ResourceName, ResourceCost, ResourceQuantity, ResourceRequiredQuantity)

Object Analysis

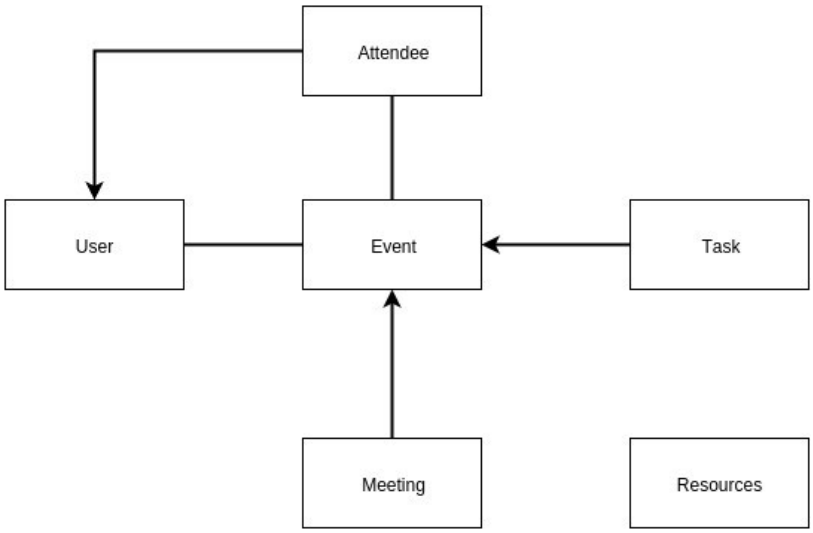
Object Listing

- Meeting
- Task
- User

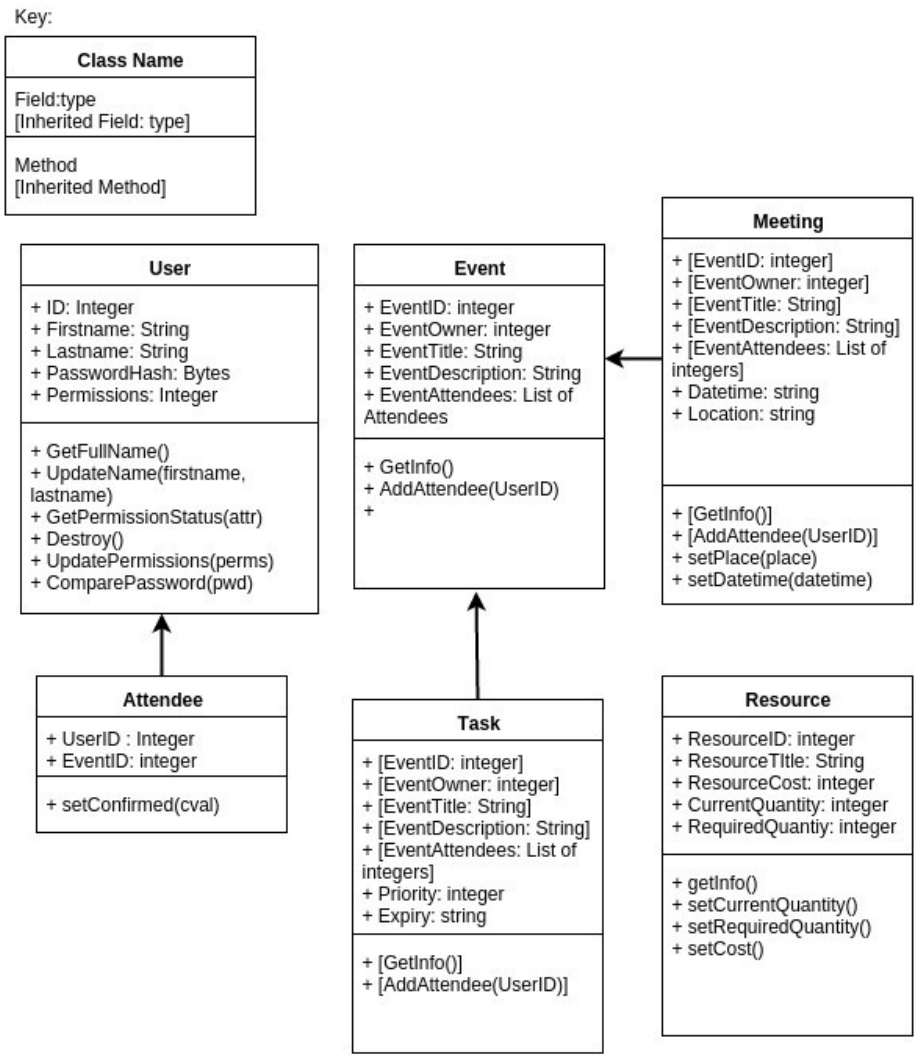
Peter East – 6303 – COMP4 Coursework

- Resources

Relationship Diagrams



Class Definitions



Other Abstractions

Constraints

Hardware

The Church currently uses a variety of low-spec PCs and Apple iMacs, the proposed system does not require and particularly high performance hardware to execute it's various tasks, however it does have some dependencies which might not be present on older computers.

The lowest specification compute that is still used in the Church has these specifications:

- 15” 4:3 Display
- Intel® Pentium 2.13GHz (Mid 2009)
- 2Gib DDR2 Ram
- 100MBit NIC
- 60GB Hard Disk Drive
- Integrated Graphics

The proposed system will not require computational resources beyond this computer. Maybe when the database gets large, the users may notice a slight delay in the computer accessing the database, but the effects should be negligible.

The staff team uses a variety of Laptops and Desktops all connected to a central server which shares some files and resources between the different computers, the database file for the proposed system is going to be stored on this server so the users will have to be connected to the Church's local network for the proposed system to work.

Software

Some of the users of the system are not trained to use computers and may be disorientated by radical changes to a computer system such as changing the operating system, however Python, and the PyQt library that the proposed system will be using are cross platform and will work on Windows, Mac and Linux.

Time

The only deadline for this software is the January 2016 deadline set by my teacher. There is no rush as far as the Church is concerned.

User Knowledge

None of the members of the staff team have qualifications in ICT or computing related subjects and there is no corporate scheme to train the staff team. Beyond basic word processing and dealing with emails, the staff team has little or no knowledge of computers which is why it's essential for the software to be as familiar and logical as possible, as well as being supplied with a full user manual.

Access Restrictions

Each user of the system needs their own section that contains all of their data, which will be password protected.

Due to the sensitive nature of much of the data held within the system, the administration will have to carefully consider how they will comply with the data protection act with the storage of their database file.

Limitations

Areas to be included in future computerisation

An extension to the system could be developed to replace the checkout element of the cafe which would automatically update resources and it would be able to keep financial records and report them to the cafe management on a regular basis, this would completely cut out the time required to input the updates to the sales information at the end of the day, and it would give the management live insights into the operation of the cafe.

Solutions

Alternative Solutions

Solution	Advantages	Disadvantages
Custom set of spreadsheets	Does not require bespoke software	does not store data securely, does not organise data, difficult to share information between people. No support for notifications
“Webapp”	Cloud resource, accessible anywhere from any device, off-site backups & server resources. Support can be issued remotely	Has a regular service charge, a web based application is open to everyone in the world therefore the application must be secure.
Revising the current system	Very low cost, no external contractors required, no need to retrain and/or learn new skills.	All of the current problems will still exist, management of data becomes a manual and laborious task.
Command Line application (CLA)	Quicker and easier to program, most storage and processing efficient solution	Requires significant training and documentation. Most of the users working for the client have little or no computer experience therefore a command line application would be completely foreign to them and would probably scare them away from using it.
Desktop application with GUI	Can be written in Python so all of the core code will be the same as in a CLA, except it will have a GUI to operate those functions. Layout can be easy to use and can include easy to reach help at every point of	GUIs are more time consuming to program than CLAs as the layout of the UI requires significant design process. Also rendering and operating GUI requires more processing than a CLA.

Solution	Advantages	Disadvantages
	entry. Could be used with a touch screen for ease of use.	

I have chosen a python GUI desktop application because:

- The application will meet my client's specific needs in a user friendly way that cannot accidentally be changed, unlike a spreadsheet.
- The digital storage of the user's data will fit on existing hardware.
- All the data contained in the database can be easily backed up and restored.
- The GUI has all of the advanced features of the CLA but they are more easily accessible.
- The python language has a balance programming simplicity and computing versatility that make it perfect applications such as these, when there's a relatively small time frame but the task requires some advanced features.

Design

Short Description of the Main Parts of the System

The system contains three main elements: The subsystem for managing meetings, the subsystem for managing referral tickets and the subsystem for managing resources and accounting. The meeting management subsystem will contain a record of all the meetings for each member of the staff team, therefore as part of the global system, there will be a representation of the staff team and anyone who might be involved in any meetings. This subsystem will also be responsible for the reminding the users of their meetings and informing users when they've been invited to a meeting. The next subsystem is the system for managing support and referral tickets, this will automatically pass the ticket on to whoever is on the rota to deal with that ticket at the given time. The tickets will be listed in order of priority, which is set when they're submitted, the priority will increase with time to ensure that nothing is ignored for too long as often the issues that would be reported are very time sensitive. The tickets will be kept securely in an encrypted section of the database, and each individual ticket will only be accessible by people for whom it is relevant to ensure confidentiality and compliance with various laws concerning such information. The third subsystem, designed for managing the material resources within the Church will consist of a record of all of the finite resources that the Church regularly purchases and sells, the subsystem will also keep track of the money throughput. Because many of the people who would operate this system will not be trained nor contractually obliged to give a satisfactory quality of service, the security and access rights components of the system are of paramount importance, not only to prevent breaches in confidentiality but also to ensure that nobody is confused when the interface is more complicated than necessary therefore the system needs to determine which parts of the system are relevant to a particular person and only show them those parts.

Flowchart showing overview of the entire system

User Interface Design