Contents

[1.01 - Background information and problem identification 2](#_Toc440578048)

[1.02 - Interviews with Peter Jones (one of the main clients) 4](#_Toc440578049)

[1.03 - Current system 5](#_Toc440578050)

[1.04 - Identification of the prospective users 7](#_Toc440578051)

[1.05 - Acceptable user needs and limitations 8](#_Toc440578052)

[1.06 - Data sources and destinations 9](#_Toc440578053)

[1.07 - Data volumes 13](#_Toc440578054)

[1.08 - Analysis data dictionary 15](#_Toc440578055)

[1.09 - Data Flow Diagrams (before and after) 18](#_Toc440578056)

[1.10 - Objectives 22](#_Toc440578057)

[1.11 - Realistic appraisal of the feasibility of potential solutions 24](#_Toc440578058)

[1.12 - Justification of chosen solution 27](#_Toc440578059)

[2.01 IPSO Chart 29](#_Toc440578060)

[2.02 Overview of Proposed System 29](#_Toc440578061)

[2.03 System Flowcharts 31](#_Toc440578062)

[2.04 Minimum Hardware Requirements 38](#_Toc440578063)

[2.05 Description of Modular System 38](#_Toc440578064)

[2.06 Data Dictionaries 43](#_Toc440578065)

[2.07 Database Normalization 49](#_Toc440578066)

[2.08 ER Diagram 51](#_Toc440578067)

[2.09 Identification of appropriate storage media 52](#_Toc440578068)

[2.10 Identification of algorithms using structured English or pseudo code 55](#_Toc440578069)

[2.11 Description of needed queries using SQL 57](#_Toc440578070)

[2.12 Screen Designs 59](#_Toc440578071)

[2.13 Security and Integrity of Data 84](#_Toc440578072)

1.01 - Background information and problem identification

City Church Grimsby was previously two Churches, The Vine and New Life Grimsby. However, on 23rd February 2009 these two Churches merged together and became what it is known today as – City Church Grimsby. At the start of the merge it was a congregation of about 100 people but has since grown to 200 regular attendees. City Church host a Sunday morn­­ing service almost every Sunday of the year. There are also other groups throughout the week taking place. The Sunday morning service operates from Lisle Marsden School, which is a building that they rent. This is because the building that they are based is not big enough for the size of the congregation that they have, so they choose to rent out the school each Sunday morning from 9-1. The pastor of the Church is David Jones who works full time within the Church. He has multiple responsibilities within the Church.

The people who organise the rotas are Rebecca Gallant, Peter Jones and Joy Wooffindin. They all create individual rotas and email them out to volunteers included on the rota each week. There are around 44 volunteers that work for City Church Grimsby and about 6 people that are on a paid salary. Their main office is based at 251-259 Freeman Street, Grimsby.

The 200 regular attendees come to the Church most Sunday mornings and will take part in the Churches service. However out of the 200 regular attendees, there are about 25 volunteers making sure that the service runs to plan on at any given time. All of the 44 volunteers/workers will rotate every week to ensure that most people are happy with how often they are volunteering their service. Becky Gallant is in charge of the main rota, Peter Jones is in charge of the Technical and Worship rota and Joy Wooffindin creates the CCKids rota.

There are multiple jobs that take place at City Church. The table below lists all of the jobs and the roles that they play on a Sunday morning service:

|  |  |
| --- | --- |
| Job | Job Role |
| Guitar | Plays acoustic guitar during worship |
| Lead Vocals | Is the main singer during worship |
| Keyboard | Plays keyboard during worship |
| Vocals | Does background vocals during worship |
| Percussion | Plays a percussion instrument (Drums, Cajon) |
| PA | Public address system is where sound is configured and is usually run by two people |
| Bass | Plays bass guitar during worship |
| Laptop | Changes lyrics and slideshow accordingly |
| Transport | Transports all the equipment to and from Lisle Marsden |
| Welcome | Welcomes people as they enter the Church. Give them a newsletter |
| Welcome/After service Coffee | Serves coffee pre and post service |
| Communion | Serves communion |
| Security | Looks after the site during the service |
| Offering Counting | Counts the offering |
| Prayer Co-Ordinator | Sets up prayer before, during and after the service |
| Set Up | Sets up the Church ready for the service |
| Set Down | Returns the school back to how they found it |
| TOTS | Volunteering in Tots section |
| Kids | Volunteering in Kids section |
| Tweens 1 | Volunteering in Tweens 1 section |
| Tweens 2 | Volunteering in Tweens 2 section |

The people that fulfil the roles in the table above will rotate according to availability and how often they wish to be of service. Also some people will have multiple jobs, but they will do these jobs on different weeks. This means that someone could be on coffee one week, but then be on welcome the next week.

There are three rotas for the different sections of the Church. Three different people make their rota and distribute it to certain sections of the voluntary team at the Church. This is so people can see what they are on, in which week, without having to look through all of the rotas. The first rota is the general running of the Church and is managed by Becky Gallant. The second is worship and technical, which is run by Peter Jones. And finally the last one is the CCKids rota, which is run by Joy Wooffindin. The rotas are completed by Becky, Peter and Joy on a regular basis and are put into a Spreadsheet before being emailed out to the relevant volunteers accordingly. If anyone wants to change the rota they will have to speak to the corresponding person, e.g. Peter Jones, who will change it by swapping that person with someone else. The updated Spreadsheet would then get emailed out to everyone again. Copies of the Spreadsheets are also kept on dropbox for people to access. However, anyone with permission to do so can change them, which can cause issues.

The current way of producing rotas can be time consuming. Also it is difficult to plan around people and their availability. Each time a change to the rota is necessary, they have to modify the current rota and then email it out to everyone again. Also there is no reminder system in place so people may forget if they are volunteering that week causing issues as the jobs that have been allocated to them will not get done.

1.02 - Interviews with Peter Jones (one of the main clients)

I have met up with the client multiple times and interviewed the user over a four-week period. Below are the main points that I have gathered from these interviews.

What is your current system for creating a rota?

Three people create rotas using Microsoft Spreadsheet or Microsoft Word. There is a pre-made template that they will edit and use every month to save time in creating the rota. We will then email it out to all the volunteers that need it to be emailed out to from our personal email address. This applies to Rebecca, Joy and I.

What are the good aspects of the current system?

The rotas ensure that everything will be organised and everyone will have a set job to do on that day. This means that everything will go as smoothly as possible. Also the template is good so we don’t have to create a new one every time we go to make a new Spreadsheet.

What are the bad aspects of the current system?

It is very time consuming so it will take a lot of time to create a suitable rota for the month. Also there is no formal way to tell us the dates that people are not available so we cannot be sure when people are not available.

What are the specific needs you want from the new system?

We want an automated rota that will create a rota for us whilst taking into account the days that people are not available. We would also like for it to be automatically emailed out to the relevant people and then again a few days before they are scheduled on the rota, if possible.

How many people will be using the new system?

There are 3 main people that will be using the software to create the rota and about 50 people who will receive the emails.

Are there any features that you would like to keep?

I like that we can use a template each time so that we can save time by using one that we have already made earlier. This also helps the volunteers that view the timetable each week as they can easily read it. However, we are open to a new timetable.

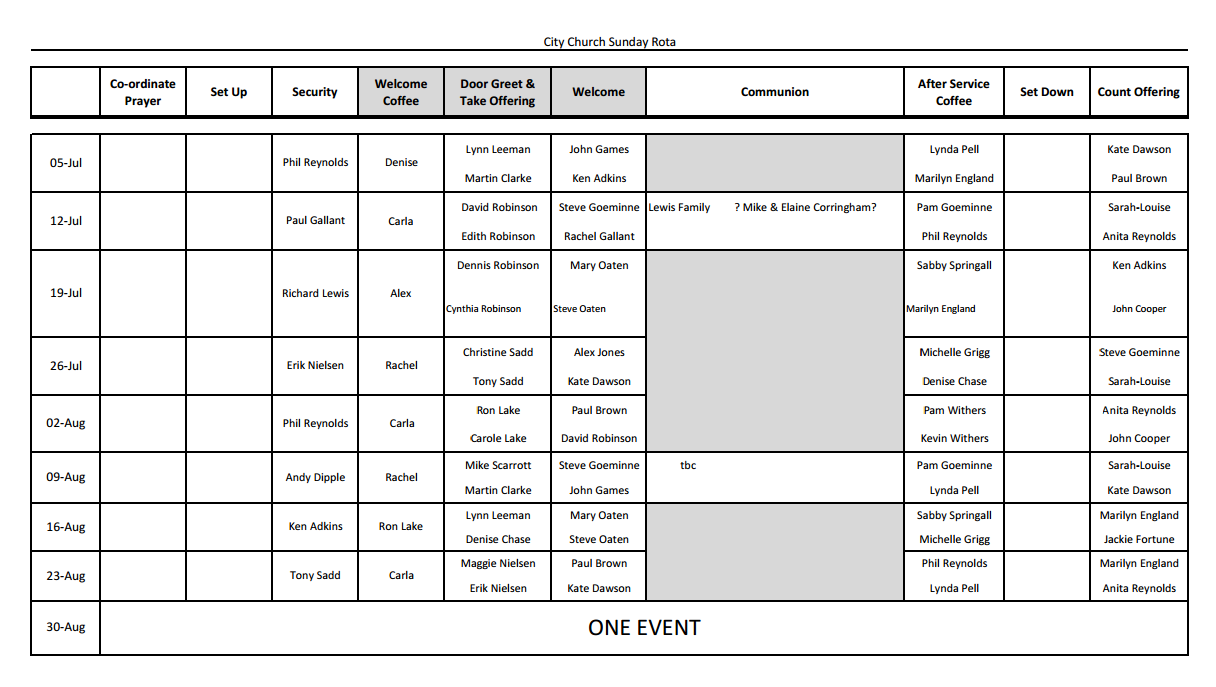
Would you like the program to work for all of the rotas?

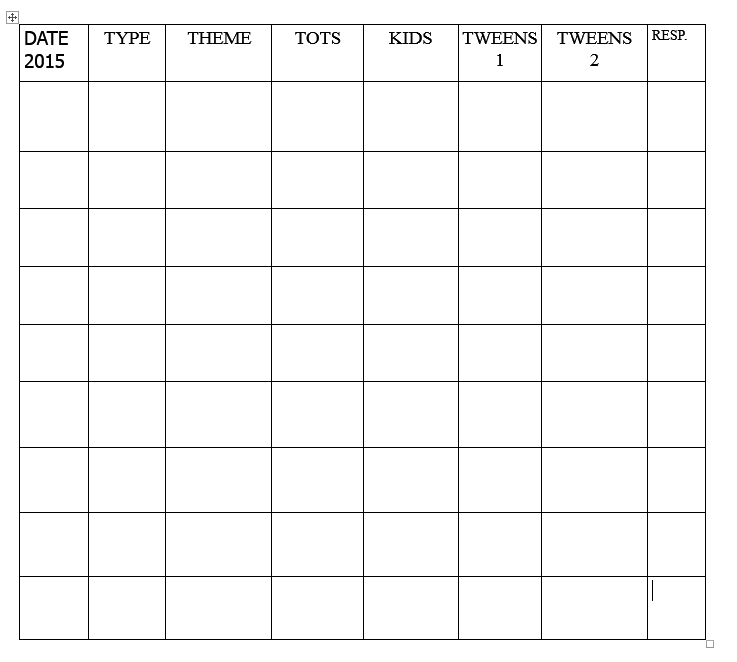
Yes, we would like to use the same system to create the rotas. We want it to work for the general, Sunday school and technical rota.

1.03 - Current system

In the current system the clients will first get their blank Spreadsheet templates on their computer like the template below. They will then refer to any information that any of the volunteers have provided about their availability on certain dates. The volunteers can request not to be included on the rota on certain days in many different ways. This means that it can be very confusing for the person that is organising the rota as they will have received lots information on when people are not available which means that the current system is prone to errors, which will also take more time to complete the rota. They will then input the job that the volunteer will have to the corresponding week so that the volunteer knows what they are doing. This will then be repeated for the number of weeks that they wish to plan ahead for.







The Spreadsheets will be then uploaded to dropbox so that everyone can access it even if they don’t receive the email. Also, because it was uploaded to a cloud service, they allow certain people to edit it if they have the correct permissions. The people with the correct permissions will be people who are more senior/have been given permission by the volunteer in charge of the rota. This is so people cannot edit it without them telling the volunteer in charge of the rota, which helps to minimize confusion. After this has taken place the clients will then input the volunteers’ email addresses one-by-one and email the rotas out to the relevant people.

After all of the planning is completed, the rotas will then be uploaded to dropbox so that relevant people from the Church can access if for any reason they don’t get the email with

the rotas attached. The dropbox files are protected with permissions so that only certain people are able to edit them. Therefore, there is no confusion with dates being changed without first being approved by the person in charge of the rota or any accidental mistakes.

After this has taken place they will compose a new email from their current email address and will have a pre-set group of people that they will email the rota to so they just use the pre-set email address and create an email to the people notifying them that the new rota is attached and that it has been uploaded to dropbox. However, if they wish to download and view it, they will have to have a Spreadsheet processing application in order to view it. This is a problem as some people don’t have one installed.

Once the email is sent out, the volunteers will all view the rota and double check it is okay. If it is then the volunteer should turn up on the correct date that they have been allocated. However, in the eventuality that they cannot now complete their allocated role or there is an error with the Spreadsheet, then they would then have to contact the person that is in charge of the rota. They will normally contact them by email, text or in person.

Once the person in charge of the rota receives the email, they then correct it according to the information given to them. The Spreadsheet will then be updated. There could be multiple changes which could create more errors which could result in more complaint emails if the person in charge of the rota forgets one of the dates that someone is not available.

The problems with the current rota system are:

* It’s time consuming. Creating a rota each month takes time
* There is no standardised way people can say they are not available
* People don’t get reminded so people forget to fulfil their volunteering duty
* You have to guess where to put people on the rota as you have to plan around them which could result in some people involved in volunteering duties more than others
* There is no formal way to tell the person in charge of the rota that you are unavailable on a certain date
* The rota is very prone to errors as the person in charge of the rota may forget or not see a mistake
* People may not know how to get in contact with the person in charge of the rota to tell them when they are not available.

1.04 - Identification of the prospective users

There are going to be 3 main users of the program that will be made. The 3 main users will be Peter Jones, Rebecca Gallant and Joy Wooffindin. Pete will create the rota for the PA staff/musicians, Rebecca will create the rota for the main Church and Joy will create the rota for CC Kids. They will all use the same program. There will be slight variations with the program that they will use, however they will still be using the same program. A rota will need to be created that will suit the needs of all the volunteers while having enough volunteers to be able to run the Church. The 3 main users of the program will be responsible in the whole for creating and organising the Church Sunday morning rotas, which includes information on which volunteer will be doing what job on what date.

The volunteers will then receive an email with the full rota attached for the section(s) they volunteer in. They will then tell the main users if there are any dates they are not available. This will then be sorted so they are not doing their job on that day. They will also be put on a day they are available to volunteer. There are around 44 volunteers. Their duty will be to turn up on the date that they have been allocated and then to go do the role they are expected to do.

I have also been told there will be other people that will help organise things, but that will only be on the day that the Church service is taking place

1.05 - Acceptable user needs and limitations

When I met with the client for the second time about what he wanted the new program to do he gave me these four points:

1. Have a central rota, which a number of users can input into, from their specific areas of Church responsibility.
2. Be able to rota volunteers based on their availability or unavailability.
3. Send e-mails or text (if this can be linked up) reminders to volunteers that they are serving on the upcoming Sunday.
4. If possible, allow volunteers to e-mail in their availability or unavailability that links direct to the program, saving us having to input the data.

We then further discussed this and I also suggested that the rota should be scheduled so that people are only put on parts of the rota that they are trained in, helping everything to run smoothly.

As I researched into what the clients’ needs were, I found out that all 3clients want a similar program with the slight variation of a different template and different people and jobs. However, all 3 will still go through the same basic process to create the rota. The only tweaks will be with the rota layout and the databases that they use. Therefore, I only needed to interview one of the clients because they do meet regularly to discuss what they want from the program and were happy to give the overall responsibility to Peter Jones. Therefore, Peter is the main client I have interviewed.

Peter has also said that the all users have the capability to use a computer to a level that will allow them to access and use a program that is as complex as what the suggested new program would be. Therefore, the users of the program should not be a limitation.

The users’ needs are as follows:

* Volunteers – Be able to add and edit volunteers’ data in the database from inside the program. This will allow the users to easily use the program and the main users to update the database, which is the best way to do it as some of the users will not be able to edit the database. Also they can add unavailability to the database from inside the program.
* Rota – Once you run the program you must be able to first of all load up the correct template of the rota. It must then be able to auto-generate a rota using variables like volunteer’s unavailability and jobs they are trained and qualified to do. This will allow the program to create a suitable rota. It must then be editable. The users can then save and print the rota from inside the program.

And the users’ limits are:

* Hardware and software – I am limited by the software because of my knowledge of form based programming due to the time I have. Also I am limited by the type of software that I can use and to the type of computer that they have at the Church.
* My knowledge of VB.net – Due to me only just being introduced to form based programming on VB.net and the time constraints on. Therefore, the complexity of the problem has to be limited. Consequently, I could not do things that the client asked for like a cloud based service or send out email reminders
* Time constraints – The project has to be done by Easter 2015
* I am also limited to the operating system that I can make it work on due to my knowledge of the program. This means that I cannot be 100% sure that it can run on Pete’s Mac running OS X 10(Yosemite); however, I can try.

1.06 - Data sources and destinations

|  |  |  |  |
| --- | --- | --- | --- |
| **Existing System** | | | |
| **Name** | **Description** | **Source** | **Destination** |
| First and last name | This contains the volunteers first and last name to identify the volunteer | Spreadsheet with names, jobs and contact info | Rota which is emailed to volunteers |
| Job for service | What the job is called so it is easier for the user to identify the name instead of the job ID | Created by Peter, Joy or Rebecca | Rota which is emailed to volunteers |
| Email address | Email address of the volunteers so that they can easily see the contact details of the volunteers | Spreadsheet with names, jobs and contact info | Composition of email |
| Jobs | So the computer can easily see the Job | Knowledge of what jobs are needed | Rota which is emailed to volunteers |
| Availability | When the volunteers are available | Multiple sources | Peter, Rebecca or Joys Knowledge |

|  |  |  |  |
| --- | --- | --- | --- |
| **Proposed System** | | | |
| **Name** | **Description** | **Source** | **Destination** |
| First name | This contains the volunteer’s first name | Volunteers Database | Rota which is emailed to volunteers |
| Last name | This contains the volunteer’s last name | Volunteers Database | Rota which is emailed to volunteers |
| Volunteer ID | This is so that it is easier to identify the volunteer | Volunteers Database | Inputting into a text box in the final program |
| Email address of Volunteers | This is for the users  reference so that they  can easily set up the  email to go to the  people needed | Volunteers Database | Composition of email |
| Volunteer Phone Number | For the users’ reference so they can contact the volunteer if needed | Volunteers Database | Program |
| Volunteers’ Jobs Qualified | So the computer will know when not to put a volunteer in the rota | Volunteers Database | Program Cache |
| Volunteer DOB | So the program will automatically not schedule in the volunteer on their birthday | Volunteers Database | Program Cache |
| Jobs | List of jobs that are needed to be done | Jobs Database | Rota which is emailed to volunteers |
| Job ID | So the computer can easily access the job | Jobs Database | Program cache |
| Unavailability | So the computer will know when not to put volunteers in the rota | Unavailability Database | Program cache |
| Max Time Volunteer on in a month | So the computer can make the rota as accurately and make it tailored to the users’ needs as much as possible | Unavailability Database | Program cache |
| Rota Name | The name of the rota | Rota Database | Rota which is emailed to volunteers |
| Rota ID | The name of the rota recognizable to the program | Rota Database | Program Cache |
| Rota Template Address | So the computer will know where to load the database from | Rota Database | Program Cache |
| No. of times on | How many times a volunteer has been on this year | Statistics Database | Diagram |
| No. of times done job | More in depth than No. Of Times on. It will show how many times they have done a specific job | Statistics Database | Diagram |
| No. Of Days Unavailable | Shows how many days that they have been unavailable for | Statistics Database | Diagram |
| Average Jobs Qualified for | Shows how many jobs that the average volunteer can do | Statistics Database | Diagram |
| Number of New People per Year | Shows how many people have become volunteers in the last year | Statistics Database | Diagram |

Each of the 3 users (Peter, Rebecca and Joy) will enter all of the information to do with each of the volunteers they are in charge of using the program. Each bit of data entered is in the IPSO chart below. This will then be stored on the computer in a database; the client has requested that it all be done in ‘the cloud.’ However, I have explained to the user this may not be possible but I will try, which they understood.

|  |  |
| --- | --- |
| Input   * First Name * Last Name * Email Address of Volunteer * Phone Number of Volunteer * Volunteer Jobs Qualified For * Volunteer Date of Birth * Jobs * Unavailability * Max Time On * Rota Template Address | Process   * Login * Forgotten Password * Main Screen Selection * Modify Rota * View Rota * New Rota * Print Rota * Save Rota * Modify Volunteer * Delete Volunteer * Add Volunteer * Add Job * Delete Job * Modify Job * View Volunteer Statistics * View Whole Church Statistics |
| Storage   * Volunteer Database * Unavailability Database * Jobs Database * Statistics Database * Rota Database | Output   * Volunteer ID * Job ID * Rota Name * Rota ID * No. Of Times On * No. Of Days Unavailable * Average Jobs Qualified For * Number of New People per Year |

1.07 - Data volumes

Volunteers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label | Type | Maximum Characters | Number of bytes | Occurrence | Total bytes |
| First name | String | 20 | 20 | 50 | 1000 |
| Last Name | String | 20 | 20 | 50 | 1000 |
| Volunteer ID | Integer | 4 | 4 | 50 | 200 |
| Email Address of Volunteer | String | 255 | 255 | 50 | 12750 |
| Volunteers Phone Number | Integer | 11 | 11 | 50 | 550 |
| Volunteers Jobs Qualified for | Integer | 2 | 2 | 50 | 100 |
| Volunteer Date of Birth | Integer | 6 | 6 | 50 | 300 |
| Max Time Volunteer on in a Month | Integer | 1 | 1 | 50 | 50 |

Jobs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label | Type | Maximum Characters | Number of bytes | Occurrence | Total bytes |
| Jobs | String | 20 | 20 | 50 | 1000 |
| Job Description | String | 160 | 160 | 50 | 8000 |
| Job ID | Integer | 3 | 3 | 25 | 75 |

Unavailability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label | Type | Maximum Characters | Number of bytes | Occurrence | Total bytes |
| Unavailability | Integer | 6 | 6 | 500 | 3000 |

Rota

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label | Type | Maximum Characters | Number of bytes | Occurrence | Total bytes |
| Rota Name | String | 20 | 20 | 50 | 1000 |
| Rota ID | Integer | 1 | 1 | 5 | 5 |
| Rota Template Address | String | 500 | 500 | 1 | 500 |

Statistics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label | Type | Maximum Characters | Number of bytes | Occurrence | Total bytes |
| No. Of Times on | Integer | 5 | 5 | 50 | 250 |
| No. Of time done job | Integer | 5 | 5 | 50 | 250 |
| No. of Days Unavailable | Integer | 4 | 4 | 50 | 200 |
| Average Jobs Qualified for | Single | 2 | 2 | 50 | 100 |
| Number of New People per Year | Integer | 5 | 5 | 50 | 250 |

Estimated Storage Required

|  |  |
| --- | --- |
| Total Bytes(x) | 22580 |
| Total Kilobytes(x/1024) | 22.05 |

It is estimated by looking at current growth of the church and other factors that there will be a growth of no more than 15 volunteers in the database over the next 3 years This should have a small effect on the database, but nothing that will take up too much space. However, volunteers will come and go, which will take up storage due to there being free spaces not being filled up in the database. The amount of unused/unusable storage will depend on how the data is stored in the database. The most efficient way to store data will be used to minimise wastage on the client’s computer. Also, the program will be able to execute faster because of this.

It is also estimated that there will be 3 more jobs added in the next 3 years. However, the client has said that there shouldn’t be any jobs removed, which will have a minimal effect on storage.

The storage needed for the databases will not exceed 22kb and will not go up by a large amount. Therefore, the growth of the database will not exceed the size of the hard drive.

1.08 - Analysis data dictionary

Volunteer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Description | Size | Example Data | Validation |
| First name | String | Stores First Name of Player | 20 | John | Not Blank |
| Last Name | String | Stores Last Name of Player | 20 | Smith | Not Blank |
| Volunteer ID | Integer | Store the Volunteer ID | 4 | 0027 | Integer with exactly 4 digits |
| Email Address of Volunteer | String | Stores the active email address of volunteer | 255 | John.Smith@gmail.com | In email address format |
| Volunteers Phone Number | Integer | Stores the active phone number of volunteer | 11 | 07369453287 | Integer with exactly 11 digits |
| Volunteers Jobs Qualified for | Integer | Stores the jobs that the volunteer can do | 2 | 2 | Not Blank |
| Volunteer Date of Birth | Integer | Stores when the volunteer was born | 6 | 10/09/67 | Date format (DD/MM/YY) (Optional) |
| Max Time Volunteer on in a Month | Integer | Stores how many times a volunteer can do a month | 1 | 2 | Integer with exactly 1 digit |

Job

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Description | Size | Example Data | Validation |
| Jobs | String | Stores the name of the job | 20 | PA | Not Blank |
| Job Description | String | Gives a brief description of the select job | 170 | Sound Quality | Not Blank |
| Job ID | Integer | Stores the ID of a corresponding job | 3 | 012 | Integer with exactly 3 digits |

Unavailability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Description | Size | Example Data | Validation |
| Unavailability | Integer | Stores when a volunteer is unavailable | 6 | 18/09/15 | Date format (DD/MM/YY) |

Rota

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Description | Size | Example Data | Validation |
| Rota Name | String | Stores the name of the rota | 20 | Technical Rota | Not Blank |
| Rota ID | Integer | Stores the ID of a corresponding rota | 1 | 2 | Integer with a maximum value of 3 |
| Rota Template Address | String | Stores where the template is stored | 500 | C:\Rotas | In format of a memory address location and must be a valid location |

Statistics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Description | Size | Example Data | Validation |
| No. Of Times on | Integer | Stores the number of time a volunteer has been volunteering | 5 | 59 | Not Blank |
| No. Of time done job | Integer | Stores the amount of times a volunteer has done a specific job | 5 | 72 | Not Blank |
| No. of Days Unavailable | Integer | Stores the number of days a volunteer has been unavailable for | 4 | 12 | Not Blank |
| Average Jobs Qualified for | Single | Stores the average jobs that a volunteer is qualified for | 2 | 2.9 | Not Blank |
| Number of New People per Year | Integer | Stores the number of new volunteers each year | 5 | 9 | Not Blank |

1.09 - Data Flow Diagrams (before and after)

Existing System – Level 0



Existing System – Level 1



These two DFD’s show the flow of data of the current system and how the system will work. In the most advanced DFD (Level 1) above we can see that the volunteer will tell the user when they are not available. The user will then store this somewhere. There is no fixed way to store it as the user currently chooses where they wish to store it. After this is done, the user will create a rota from the volunteer information. As stated before, the user will get this from different sources depending on which of the three users it is. They will then output a rota usually in Spreadsheet or word format before converting it to a PDF format. The user will then create an email using the emails from the volunteer information and send it as an attachment to the relevant volunteers.

Proposed System – Level 0



Proposed System – Level 1

This is now the updated version of the system. It is a lot more advanced, yet it is more organised. First of all, the volunteer will tell the user their unavailability. The user now has a set way of storing the data. There will be a section of the program specifically for this. The user will input the volunteers name and their unavailability. This will then be stored in the Unavailability Database ready for when the rota is created.

There will also be another section of the program which will consist of adding, editing and removing volunteers. In the adding volunteers, the user can add a volunteer to the database but only if they have the required fields. This will then be added to the volunteers’ database and will be saved ready for when it is required. When the user wishes to edit a volunteer, they can select edit and then save and edit any of the data that they wish. This is also where the user will view the volunteers’ information. Finally, there is a remove section where the user can choose to remove a volunteer. To do this, the user will simply need to input the volunteers first and last name and then it will remove them from the database.

Next the user will be able to add, edit and remove jobs. This is very similar to the volunteers apart from it will update the jobs database.

The main part of the rota is where the rota is generated. The user will input into the program. When it is initiated the program will load data from the unavailability, jobs and volunteers’ database. It will then use all of the variables from these and then create a rota that will be suitable. After this is done, the user will be able to modify the rota using any drop down boxes. After this the user can save or print it. Either way it will output the rota to the user. This will then be distributed to all of the volunteers in the means that the user finds most suitable.

After pressing save, the system will create some statistics which will be saved in the database. These statistics will be viewable as either pie or bar charts or a graph. This will help the user track statistics. This can be narrowed down into either Church statistics or specific volunteer statistics. The Church statistics will show the average jobs qualified for and also number of new people per year. The volunteer specific statistics will show: the number of times a volunteer has volunteered, the number of times the volunteer has done a specific job and the number of days the volunteer is unavailable. These will all be shown as number and can be viewed as graphs, pie and bar charts.

1.10 - Objectives

1. The system must be able to hold all records of volunteers (50) and all of their information. Their information consists of; full name, email address, phone number, ID, and also can contain DOB but this is optional. It must be able to do this for however long they decided to run the program for.
2. From starting the application to the emails being sent out must take under 45 minutes, with minimal changes to the rota. This will ensure that the rota will be efficient enough to meet the clients’ needs.
3. The program must auto generate the rota first time with no/minimal glitches taken place. It must generate it within 30 seconds of initiating the program to generate the rota. This is to ensure that the client is spending as little time as possible creating the rota. This will ensure it will be more efficient than the current system that they are using at the moment.
4. There are never any conflicts with the unavailable dates in the rota. This ensures that a program is created that will meet the client’s main need of speeding up the time taken to create a program. Also the client will not have to come back to the rota and change anything due to a fault in the program.
5. Select from 3 different templates which will consequently select a different database every time the client loads up the program. This is because the client has to create 3 different rotas due to there being 3 sections in the Church on a Sunday morning. This will also make things easier because 3 different people make the rota so now they will still be able to create a rota for each section of the Church like they do in the current system.
6. The users can make any modification to the rota after it has auto-generated or saved, so that any changes that they wish to make to the rota can be easily done using the programs interface.
7. The user will be able to add a new volunteer to the database at any time they want using the program and be instantly available to be put into a corresponding rota. They must have; full name, email address, phone number, ID, and also can contain DOB but this is optional.
8. They must also be able to edit all of volunteer’s records. This consists of edition; full name, email address, phone number, ID, DOB. They must be able to edit this at any time and the changes must happen within 5 seconds.
9. They must be able to remove any volunteer from the database at any time using only their first and last name. This must completely wipe them from the database within 5 seconds.
10. The user must be able to select from 3 different rotas when they start the program up. It must then load the rota and the rest of the program within 5 seconds. It must only load the corresponding rota.
11. The user must be able to input unavailability whenever they want to. They must be able to select from any date and it be put into the database within 5 seconds of initiating the database update.
12. The user must be able to view all of the unavailability of all of the clients whenever they wish using the program. It must display the information when asked within 5 seconds of booting the program up. There must also be a search function involved so that the client can see specific dates or people’s unavailability.
13. The user must be able to add a job to the database whenever they want. They will have to have the job name and job description to create it. It must also be automatically assigned a Job ID and added to the database in 5 seconds when initiated.
14. The user must be able to add jobs to list of jobs that a volunteer can do at any time they wish to. This must take no more than 5 seconds to update when initiated and the only information they need is clients first and last name and the job they are adding to them.
15. The program must automatically start collecting data from when the first rota is created. It must collect multiple bits of data including:

* No. of times on (volunteer specific)
* No. of time done a job (volunteer specific)
* No. of days unavailable
* Average jobs qualified for
* Number of new people per year

The user must not have to add this data manually. This must always be collected every time a new rota is created.

1. The user must also be able to search through all of the records of the volunteers (50) and see the statistics in graphs or pie and bar charts. It will display a minimum of 3 different charts. These will be automatically created and must be loaded in 10 seconds; however, this can vary depending on the number of diagrams needing to be created.
2. When the rota auto generates it must take into account how many times that a volunteer can be on in a month as it cannot go over this. This will auto generate a rota that will be more accurate and requires less adjustments from the volunteer.
3. The program must be able to save the rota to the hard drive once the rota has been created and all changes have been made. This is so that the user can easily distribute the rota to all the volunteers however they want.
4. The user must be able to come back to the rota and then make any changes that they seem necessary so that if there are any errors for any reason that they can be easily rectified.
5. The user must also have the option to print the rota once it has been created and it must be in an easy to read format so that it is similar to the layout that it was in before.

1.11 - Realistic appraisal of the feasibility of potential solutions

Current System

The clients can take multiple routes instead of choosing to use the bespoke program. The first is that they continue as they are. This means that they will use the system described in 1.02. However, they will continue to face the same problems as before.

The positives of this are that:

* There will be no further training required as the users are familiar with this process
* It will cost the client no money or time as it will be already set up
* This is a proven system which, although very time-consuming, is able to produce rotas which the volunteers are used to.

However, the negatives are:

* This system is very time consuming
* Because it heavily relies on human input, there is the potential for lots of errors such as not writing down a volunteer’s availability or forgetting who can do what jobs
* There is no formal way for volunteers to tell the client’s their unavailability
* This will only work offline and there will be no online method unless the user chooses to upload the rota to an online cloud system.

However, to continue to use this system is no longer an option simply because of the amount of time that it takes to create the rota. Therefore, this system no long suits the client’s needs.

Rota Geek

Research has been carried out on different ‘off the shelf’ software packages. One such package was found on a website called <http://www.rotageek.com/>. This is a company that have a software package that will have a data driven scheduling and an automatic scheduler that is currently in beta. The pricing of this software is £3/employee/month. This would cost the church £1800 based on the number of volunteers that they have. This is very expensive as the Church has a tight budget and, as the client has stated, there is no room in the budget to spend this sort of money on such a package.

The advantages of this system are:

* Proven system that will work well
* Constantly updating and adding new features
* If you have any problems, you are guaranteed to be contacted within 1 day by their customer service team
* There is a mobile app
* They will send out email and text alerts of any new rota’s or changes to the rota
* You can manage the rota from multiple locations
* Swap requests can be done in the program
* You will receive help from an account manager when initially getting started
* It is available online.

On the other hand, the disadvantages are:

* This is a very expensive product which the Church has not got a budget for
* The automatic scheduling is in beta so there may be many glitches
* The product is not bespoke for their system so it will not meet all of their needs
* Further training will have to take place to get used to this new system.

So for these reasons this will not be a feasible option. The client simply does not have the budget to spend on such a package.

Bespoke Software 1

After considering the above, a bespoke solution appears to be the best solution for them. The client has also said during an interview with them that bespoke software is exactly what they want. This will be custom made for them and will meet all of the client’s need. The bespoke software will take time to create but long term will save the user a lot of time. The bespoke software would be a software application that would run on their computer.

The advantages of using bespoke software 1 would be that:

* It is bespoke so this means that it should meet all of the client’s requirements
* The software will have no features that the client does not need therefore will not waste any time when using it
* The system will automatically generate the rotas, therefore saving the client time.

However, the main disadvantages of the bespoke software 1 is that:

* It will take a long time to develop a system
* There will probably be little online support
* There will be few updates after it has been released and probably no new features unless asked for
* Auto-generating could create errors which could make it slower.

Bespoke Software 2

The bespoke system listed above is not the only bespoke system that could be created as there could also be another system where rotas will be created partly by the computer and partly by human input. This means that it will do everything the same as the bespoke system 1 but it will not auto-generate a rota. However, it will still be easy to create a rota as there will be drop down boxes with only the volunteers who are available for that particular week.

The advantages of this bespoke system 2 are very similar to bespoke system 1, which are:

* It is bespoke so this means that it should meet all of the client’s requirements
* The software will have no features that the client does not need therefore will not waste any time when using it
* There will be no glitches caused by the program when creating the rota.

However, the main disadvantages of bespoke system 2 is that:

* It will take a long time to develop it
* There will probably be little online support
* There will be few updates after it has been released and probably no new features unless asked for
* Because the rotas have not been auto generated, it will be a lot more time consuming to create the rotas.

Bespoke Software 3

The final bespoke system the client could have is a hybrid between bespoke system 1, bespoke system 2 and the current system. This could involve them simply storing data in the database and viewing it through the program. This would involve all of the features of bespoke systems 1 and 2 apart from being able to creating a rota and the rota auto-generating. The client would have to create a rota using excel.

The advantages of using this would be that:

* It is bespoke so this means that it should meet all of the client’s requirements
* The software will have no features that the client does not need therefore will not waste any time when using it
* The new system will be more reliable and faster than the current system
* Don’t need to be fully trained as they already know how to use excel.

However, the disadvantages are:

* It will take a long time to develop it
* There will probably be little online support
* There will be few updates after it has been released and probably no new features unless asked for
* Because the rotas have not been auto generated, it will be a lot more time consuming to create the rotas.

Therefore, after ruling all of the other systems out, it has come down to choosing between bespoke system 1, 2 and 3. Bespoke system 1 has been chosen because it will meet all of the client’s needs even though it will be more complex. Therefore, a system will be created that is fully automated and will require minimal user input to make it as fast and efficient as possible for the client.

1.12 - Justification of chosen solution

Bespoke system1, which is fully automated, will meet all of the clients’ needs. One of the main issues with the current system is that it takes too much time to create the rotas. The bespoke system will be created with the main aim of speeding up the time it takes to create the rotas, which will in turn improve the client’s experience when using the system.

After interviewing the client, they stated that they would also like to have a statistics table with all of the statistics taken from the created rota. The statistics will include volunteer specific statistics like the amount of time people volunteer, the amount of times volunteers have done specific jobs and the total numbers of day’s volunteers are unavailable. There will also be other statistics such as the number of roles the average volunteer can do and total number of new volunteers. This will suit the client’s needs as it will allow the client to track the Church’s progress visually using visual presentations such as pie charts, graphs and bar charts.

Also the ability to add, edit and view volunteers will suit the client’s needs because this will add to the automatic rota creation system and it will make it as accurate as possible. Furthermore, it will also suit the client’s needs because the system will speed up the time in the long run to create the rota even though the initial adding of information about the new volunteers will take some time.

The client will be able to add unavailability in the program, which supports the main aim of it taking less time to create a rota because it will automatically detect when volunteers are unavailable and not allow them to be placed in that date. Also the maximum number of times the volunteer can be on in a month will make the program a lot more precise and will therefore make the creation of the rota less time consuming.

Normally this would be a very expensive system to create. Software companies would charge a lot of money to create a bespoke system for the client. Therefore, Bespoke System 1 would be a better solution for the client rather than something like Rota Geek or getting a company to create it for them as it will save them lots of money due to the system being created for free as part of a project.

Therefore, the bespoke system, which is fully automated, is the best solution for the client. This solution will be the most time efficient way for the client to create rotas and for the program to gather all of the statistics from the rotas for a very accurate and detailed report, which the client can use to their advantage.

**2.01 IPSO Chart**

The chart below shows, at the most basic level, what happens to the data in the proposed system in the forms of Input, Output, Process and Storage:

|  |  |
| --- | --- |
| Input   * First Name * Last Name * Email Address of Volunteer * Phone Number of Volunteer * Volunteer Jobs Qualified For * Volunteer Date of Birth * Jobs * Unavailability * Max Time On * Rota Template Address | Process   * Login * Forgotten Password * Main Screen Selection * Modify Rota * View Rota * New Rota * Print Rota * Save Rota * Modify Volunteer * Delete Volunteer * Add Volunteer * Add Job * Delete Job * Modify Job * View Volunteer Statistics * View Whole Church Statistics |
| Storage   * Volunteer Database * Unavailability Database * Jobs Database * Statistics Database * Rota Database | Output   * Volunteer ID * Job ID * Rota Name * Rota ID * No. Of Times On * No. Of Days Unavailable * Average Jobs Qualified For * Number of New People per Year * Rota |

**2.02 Overview of Proposed System**

The proposed system will first of all be password protected as it will be on the main church computer. It will have no user name as the client doesn’t need that level of security on this program. The user will be met with a screen that asks them for a password. They will enter the password and click log in. If the password is correct, the user will go on to the next screen. If the password is not correct it will display a message asking the user to re-enter the password and they will stay on the same screen. However, if the user cannot remember their password they can also change it after going through a few security measures.

The next screen will present the user with a list of options that they can click. One of the options is that the user will be able to access is the “Volunteers” screen. There will then be more options on this screen such as “Add Volunteers”. This will enable the user to add volunteers to the database. They will fill out a form containing the volunteer’s first and last name, email, phone number, jobs qualified to do, maximum times on in a month and date of birth. The user can also modify this data and remove a volunteer if they want to.

On the rota screen the user will be able to create, modify or view a rota. When they create a rota it will use data from the volunteers’ database like the volunteer’s name, date of birth, maximum times on and jobs qualified. The other databases that can be used are unavailability and jobs database. The user will be able create a suitable rota using all of these variables. If they wish to edit the rota, they can use a drop down box to edit any data which will also take into account all of these factors. Modifying the rota will be very similar process apart from the user will select a rota and then use the drop down boxes to modify it. If they wish to view a rota, they simply load up the correct rota and then view it in a read only format.

Unavailability will allow the user to add, modify or delete a volunteer’s unavailability. If they wish to add a volunteers’ unavailability they will simply input the volunteer’s name and then they will be able to add a volunteers’ unavailability to the database. If they wish to modify it, they will select a volunteer and then be able to modify the unavailability from the database to a different date. If they wish to delete unavailability, they can select the volunteer and the date they wish to remove and then it will delete the date they selected.

The jobs screen will allow the user to add modify or delete a job from the database. When they want to add a job, they go to the add jobs screen and will add the job name and job description and then press add and it will save the job and give it a job ID. If they wish to modify a job, they will type the job name in and then they can edit the jobs database. When they want to delete a job, they will type the job name in then press delete and it will delete it from the database.

When the user wishes to view statistics, they will either choose volunteer or church using a radio button. If they pick volunteer, they will type in the volunteer’s name and then they will be able to see the volunteer’s statistics if they exist in the database. If they pick church, then they will be able to see the whole church statistics.

When the user has finished, they press log out and then they will be logged out of the program and return to the login screen.

2.03 System Flowcharts

*Figure 2.1 – Login Screen*

The user will enter the username and password in boxes

Username + Password

Login Screen

Does the user know the password?

No

Yes

Changes password after answering security question

Change password

Main Screen

Gets username and password to check it with data entered

Validation

The user will input their unique username and password into the textboxes. It will then compare it to the data in the database to see if they have the correct credentials to access the database. If it is correct, then they will be granted access to the program. However, if it is not correct then the user will not be able to use the program and an error message will pop up to tell them their Username/Password combination is not correct.

If they wish to change their password, then they will go onto the change password screen where they will enter their username and then they will have to answer a security question correctly. If it is entered correctly, they will then be able to change the password for the selected user.

*Figure 2.2 – Main Screen*

Main Screen

Rota Selection

Volunteers Selection

Jobs Selection

Log Out

Statistics

Unavailability Selection

This is the main hub of the program. Everything will be accessed through this screen.

*Figure 2.3 – Rota Screen*

Back to main screen

Rota

What do you want to do?

Create a Rota

View Rota

Modify a rota

The user will have a rota generated for them. They can then modify this rota.

Create Rota

The user will be able to modify, view or create a rota. The user will simply choose the variables. If they wish to create a rota, then they will have to get input from the variables and when they are done they will output a rota. When modifying a rota, they will input and get outputs from the rota and they will get an input from the variables. When viewing a rota, they will get an output from the rota file.

The user will be able to view a rota and only view a rota on this screen

View Rota

The program will load up the rota from the hard drive

Rota File

The rota will be loaded up and then can be modify.

Modify Rota

All the variables across the unavailability and volunteers’ database will be inputted into the program

Variables

Back to main screen

What do you want to do?

Unavailability

*Figure 2.4 – Unavailability Screen*

Add Unavailability

Delete Unavailability

You will be able to remove any unavailability previously inputted into the database

Delete Unavailability

This will allow the user to add unavailability to the database

Add Unavailability

Modify Unavailability

The user will be able to modify any of the volunteers’ unavailability on this page

Modify Unavailability

This will have all of the volunteers and all of their unavailability

Unavailability

When adding unavailability, the user will be able to only input into the unavailability database. However, when modifying or deleting, they will have an output from the unavailability database and also be able to input into it.

*Figure 2.5 – Jobs Screen*

Add Job

When adding jobs, the user will be able to only input into the jobs database. However, when modifying or deleting, they will have an output from the jobs database and also be able to input into it.

Jobs

Back to main screen

This will have all of the jobs stored

Jobs

The user can select a job and then remove it from the database

Delete Job

The user will be able to load up all the information on a job and then modify that job

Modify Job

This will be where the user will add a job to the database.

Add Job

Delete Job

Modify Job

What do you want to do?

*Figure 2.6 – Volunteers Screen*

Add Volunteer

When adding a volunteer, the user will be able to only input into the volunteer database. However, when modifying or deleting, they will have an output from the volunteer database and also be able to input into it.

This will have all of the volunteers’ data stored

Volunteer

The user can select a volunteer and then remove it from the database

Delete Volunteer

The user will be able to load up all the information on a volunteer and then modify that volunteer

Modify Volunteer

This will be where the user will add a volunteer to the database.

Add Volunteer

Volunteers

What do you want to do?

Back to main screen

Modify Volunteer

Delete Volunteer

*Figure 2.7 - Statistics Screen*

When viewing statistics, the user will only be able to see the statistics. They will not be able to modify any of the statistics

This will have all of the statistics data is stored

Statistics

This is where you will view the statistics of the church or the selected volunteer

View Statistics

Statistics

Back to main screen

2.04 Minimum Hardware Requirements

1Gigahertz (GHz) or faster processor

VGA 640x480 or higher-resolution screen supported by Microsoft Windows.

Microsoft Windows 95 or later.

1Gb of RAM

4GB of hard disk space

The user will also need input devices in the form of a keyboard and mouse. The keyboard must be in the QWERTY format and it is recommended that it is an English(UK) keyboard. The mouse must be a standard mouse with a left and right click. There must also be a monitor with a monitor with a minimum resolution of 720\*1280 to be able to display the program. It is also optional that the user has a printer that is connected to the computer so that they can print off the rota if they wish to.

**2.05 Description of Modular System**

*Figure 3.1– Login Screen*

1. Login Screen

2.Main Screen

1. Forgotten Password

1.2. Change Password

* 1. Security Question

Figure 2.1 shows the login screen. The user will be able to login to be able to use the program, or they will be able to change their password if they have forgotten what their password is.

If the users do not know the password that they use to login, they will have to go through this screen to be able to change their password to enable them to login and then use the program.

This will ask the user for their username to be able to load up the security question. Then they will have to answer the security question correctly. They just enter the username that they use to login and then it will take them to the next screen.

When the user has entered their security question correctly, they will then have the option to change their password. They will enter the password they wish to have twice so that they are sure that it is the password that they wish to have.

This will take the user to the main screen after they have entered their username and password correctly.

f*igure 3.2– Main Screen*

6. Logout

5. Statistics

4. Jobs

3. Volunteers

1. Unavailability
2. Rota
3. Main Screen

This is the main hub of the entire program. The user can access anything from this screen. Six different screens can be accessed from this hub. The user can go to the rota, unavailability, volunteers, jobs and statistics screens. If they wish to log out of the program, they simply press the logout button.

1. This button will take you to the rota screen.
2. This button will take you to the unavailability screen.
3. This button will take you to the volunteers’ screen.
4. This button will take you to the jobs screen.
5. This button will take you to the statistics screen.
6. This button will log you out of the program and take you back to the login screen.

*Figure 3.3 – Rota Screen*

1. Rota
2. View Rota
3. Modify Rota
4. New Rota

3.1. Save

3.2. Print

3.1. Save

3.1. Save

3.2. Print

3.2. Print

The user will be able to create a new rota using this button. It will generate a rota using the databases.

This will save the rota to a selected place on the user’s storage device for use later.

When the user presses this button, they will be able to print off their rota using a wizard to use as a physical copy.

2. This will allow the user to modify a rota that they have selected to load up. They can then use the program to modify any of the data in the rota.

2.1This will save the rota to a selected place on the user’s storage device for use

later.

2.2When the user presses this button; they will be able to print off their rota using a

wizard to use as a physical copy.

1. The user can view a rota that you have selected in this screen.

1.1This will save the rota to a selected place on the user’s storage device for use

later.

1.2When the user presses this button; they will be able to print off their rota using a

wizard to use as a physical copy.

*Figure 3.4 – Unavailability Screen*

1. Unavailability
2. Add Unavailability
3. Modify Unavailability
4. Delete Unavailability

1.1 Save

2.1 Save

3.1 Save

This is where the user will add a volunteer’s unavailability. They will fill in the selected fields and then it will be added to the database.

This is the button that will upload the data from the text boxes to the unavailability database.

1. The user can modify any of the volunteer’s unavailability that they have previously uploaded to the database. They just select the volunteer and then it will load up the selected volunteer’s unavailability where they can change it to a different date.
2. This will update the selected volunteer’s unavailability.
3. The user can delete any of the volunteer’s unavailability that they have previously added to the database. They just do the same as modify apart from they don’t add another date; they just press delete.

3.1This will delete the selected volunteers, selected unavailability from the database.

1. Job
2. Add Job
3. Modify Job
4. Delete Job

1.1 Save

2.1 Save

3.1 Save

*Figure 3.5 - Unavailability*

1. This is where the user will add your job. They will fill in the selected fields and then it will be added to the database.

1.1 This is the button that will upload the data from the text boxes to the jobs database.

2. The user can modify any of the jobs that they have previously uploaded to the database. They just select the job and then it will load up the selected job to the database.

2.1 This will update the selected job.

3. The user can delete any of the jobs that they have previously added to the database. They just type in the name of the job and then press delete.

3.1This will delete job from the database.

*Figure 3.6 – Volunteer Screen*

1. Volunteer
2. Add Volunteer
3. Delete Volunteer
4. Modify Volunteer

2.1 Save

3.1 Save

4.1 Save

2.This is where the user will add a volunteer. They will fill in the selected fields and then it will be added to the database.

2.1 This is the button that will upload the data from the text boxes to the volunteer database

3. The user can modify any of the volunteers that they have previously uploaded to the database. They just select the volunteer and then it will load up the selected volunteer to the form.

3.1 This will update the selected volunteer.

4. The user can delete any of the volunteers that they have previously added to the database. They just type in the name of the volunteer and then press delete.

4.1 This will delete the volunteer from the database.

*Figure 3.7 – Statics Screen*

1. Statistics

2. Church Statistics

1. Volunteer Statistics
2. The user can view selected volunteer statistics from this.
3. The user can see the whole church statistics in this menu.

2.06 Data Dictionaries

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Name | Description | Additional Information | Mandatory | Unique | Default Value | | Type | |
| Volunteers | | | | | | | | |
| volunteerFirstName | This is the first name of one of the volunteers. | Multiple people can have the same first name. This is no optional so a check will be made to ensure that this box is filled in with characters. This will be so the user can easily select volunteers within the program. If this was not in place the user would have to memorize which volunteer has what volunteer ID. | Yes | No | Null | String | |
| volunteerLastName | This is the last name of one of the volunteers. | Multiple people can have the same last name. This is no optional so a check will be made to ensure that this box is filled in with characters. This will be so the user can easily select volunteers within the program. If this was not in place the user would have to memorize which volunteer has what volunteer ID. | Yes | No | Null | String | |
| volunteerID | This Is the unique number that the volunteer will have | This has to be unique and it will be assigned to the volunteers by the computer in the next free space. It will a four-digit number. | Yes | Yes | Null | Integer | |
| volunteerEmailAddress | This is the volunteers email address. It will be used for contacting the volunteer | Multiple people can have the same email address as some couples may want to be emailed to one account to make it easier for them. It will also have to be in the format of an email address so a check will be done to make sure that it is correct. | Yes | No | Null | String | |
| volunteerPhoneNumber | This is the volunteers phone number. It will be used for contacting the volunteer | Multiple people can have the same email address as some couples may want to be emailed to one account to make it easier for them. It will have to be an integer with exactly 11 digits | No | No | Null | Integer | |
| volunteerJobsQualified | This is a list of the jobs that the volunteer can do | This cannot be blank and can have more than one entry. The job must be a string and must be in the jobs database | Yes | No | Null | String | |
| volunteerBirthDate | This is when the volunteer was born. It is so they aren’t on the rota on their birthday | This is an optional field and must be in the format of a date so a validation check will be performed to make sure that it is correct. | No | No | Null | String | |
| volunteerMaxTimesOn | This is so that the volunteer is not on too often in a month. | This has to be an integer no bigger than 5 | No | No | Null | Integer | |
| Jobs | | | | | | | |
| jobName | This is the name of the job | This will make it easier for the user to select a job so they don’t have to remember lots of numbers. It must be characters only. | Yes | No | Null | String | |
| jobDescription | This gives the users an idea of what the job involves | This is for the users use only | No | Yes | Null | String | |
| jobID | This Is the unique number that the job will have | This has to be unique and it will be assigned to the job by the computer in the next free space. It will a three-digit number. | Yes | Yes | Null | Integer | |
| Unavailability | | | | | | | |
| unavailabilityDate | This is the date that the selected volunteer is unavailability | This has to be in the date format. Multiple people can be unavailable on the same date | Yes | No | Null | String | |
| Rota | | | | | | | |
| rotaName | This is the name of the rota. It is so it can be easily identified by the user. | This has to be a string no longer than 30 characters. There is no way to edit it in the program so you have to do it manually. | Yes | Yes | Null | String | |
| rotaID | This is the unique ID of the rota so it is easily identifiable by the program. | This has to be unique and it will be assigned to the rota by the computer in the next free space. It will a single digit number. | Yes | Yes | Null | Integer | |
| rotaTemplateAddress | This is the directory of the rota. | This is so that the computer can find where the rota is and load it up. It must be an actual location so the program must validate it to make sure that it does in fact exist and it is in the correct format. | Yes | Yes | Null | String | |
| vStatistics | | | | | | | |
| vStatisticsNoTimesOn | This is a record of a specific volunteer and the times that they have been on | This is an integer that can have any value. This value is automatically added by the computer | Yes | No | Null | Integer | |
| vStatisticsNoOfTimesDoneJob | This is a volunteer and job specific record that is similar to noTimesOn. | This is an integer that can have any value. This value is automatically added by the computer | Yes | No | Null | Integer | |
| vStatisticsNoOfDayUnavailable | This will record the number of days that a volunteer is not available | This is a counter for the amount of time that a selected volunteer has been unavailable. | Yes | No | Null | Integer | |
| cStatistics | | | | | | | |
| cStatisticsAverageJobsQualifiedFor | This is the amount of jobs that a volunteer is qualified for on average | This is a number that will be calculated by the database. | Yes | No | Null | Single | |
| cStatisticsNumebrOfNewPeoplePerYear | This is the number of volunteers added in the last 365 days | This is a counter that will be calculated by the database and will require no user input | Yes | No | Null | Single | |
| login | | | | | | | |
| loginUsername | This is the username that the user will use to log in | This can only be characters and cannot be longer than 15 characters. Will be used on the login screen | Yes | Yes | Null | String | |
| loginPassword | This is the password that will correspond to the username | Cannot be longer than 20 characters. Will be used on the login screen | Yes | Yes | Null | String | |

2.07 Database Normalization

|  |  |  |
| --- | --- | --- |
| UNF | 2NF | 3NF |
| volunteerFirstName | \*volunteerID\* | \*volunteerID\* |
| volunteerLastName | volunteerFirstName | volunteerFirstName |
| volunteerID | volunteerLastName | volunteerLastName |
| volunteerEmailAddress | volunteerEmailAddress | volunteerEmailAddress |
| volunteerPhoneNumber | volunteerPhoneNumber | volunteerPhoneNumber |
| volunteerJobsQualified | volunteerBirthDate | volunteerBirthDate |
| volunteerBirthDate | volunteerMaxTimesOn | volunteerMaxTimesOn |
| volunteerMaxTimesOn |  |  |
|  | \*volunteerID\* | \*volunteerID\* |
|  | \*volunteerJobsQualified\* | \*volunteerJobsQualified\* |

|  |  |  |
| --- | --- | --- |
| UNF | 2NF | 3NF |
| jobName |  | jobName |
| jobID |  | jobID |
| jobDescription |  | jobDescription |

|  |  |  |
| --- | --- | --- |
| UNF | 2NF | 3NF |
| rotaName |  | rotaName |
| rotaID |  | rotaID |
| rotaTemplateAddress |  | rotaTemplateAddress |

|  |  |  |
| --- | --- | --- |
| UNF | 2NF | 3NF |
| vStatisticsNoTimesOn | vStatisticsNoTimesOn | vStatisticsNoTimesOn |
| vStatisticsNoOfTimesDoneJob | vStatisticsNoOfDayUnavailable | vStatisticsNoOfDayUnavailable |
| vStatisticsNoOfDayUnavailable | cStatisticsAverageJobsQualifiedFor | cStatisticsAverageJobsQualifiedFor |
| cStatisticsAverageJobsQualifiedFor | cStatisticsNumberOfNewPeoplePerYear | cStatisticsNumberOfNewPeoplePerYear |
| cStatisticsNumberOfNewPeoplePerYear |  |  |
|  | \*vStatisticsNoOfTimesDoneJob\* | \*vStatisticsNoOfTimesDoneJob\* |
| \*VolunteerID\* | \*VolunteerID\* | \*VolunteerID\* |

|  |  |  |
| --- | --- | --- |
| UNF | 2NF | 3NF |
| loginUsername |  | loginUsername |
| loginPassword |  | loginPassword |

2.08 ER Diagram

Unavailability

Volunteers

Rota

Jobs

Unavailability

Volunteers

Statistics

|  |  |
| --- | --- |
| Many to Many | Many - One |
| Volunteers - Statistics | Rota - Volunteers |
| Volunteers - Jobs | Rota - Jobs |
| Volunteers - Unavailability | Rota - Statistics |
| Statistics - Unavailability | Rota - Unavailability |

2.09 Identification of appropriate storage media

Introduction

There is no way of knowing exactly what the storage requirements of the proposed system will be. However, a previous estimation of the size of the database carried out, which was 22.05KB and also an additional 80 MB for the program. This is based off the size that program previously made in visual basic and then comparing this to the size of this project, which is what the estimated the size of the program will be. This will be used as the estimation of the size throughout this section (88.02MB).

**CD-ROM**

A CD-ROM is an optical disk which contains data. Its capacity is between 650-900MB and it reads and writes data at 150KB/s. What differentiates this from the other types of CD’s is the ROM part. This stands for “Read Only Memory”. This means that computers can read off the disc, however they cannot write onto the disc as it is not writable and not erasable. Up until about 10 years ago this was the go-to distribution method of software. This would be suitable as it is large enough to hold all of the data as estimated and it has substantial room for expansion. It is also portable enough to carry around, however it is not as portable as some of the storage media on this list. It should also be able to read fast enough as there is no need to access that much of the database at once. The user will not be able to modify any of the files so it shouldn’t be the user’s fault if any of the data get corrupt.

The user cannot erase data on the disc. However, this could pose as a security issue because if they want to get rid of data the only way to do this would be to destroy it and the user may not want to do this. Therefore, the data could remain on the disc for a while which means that someone could get hold of the sensitive information on the disc.

A CD-ROM would be a good method for backing up all of the data on the database as it cannot be re written. However, this would not be good for the main program and running the program because the user cannot write or erase data and they need to do this because of the database.

**DVD-ROM**

When comparing the DVD and CD they are very similar with the only real differences being the speed that it reads and writes at and the storage capacity. The capacity can range from 4.7GB-17.08Gb in size making it more than adequate to store the program on with no concerns that it will reach its storage capacity. It can also read and write at 1.32 MB/s making it so that there will be next to no waiting time for the program to read data.

This would be a great means of distributing the software due to its size and portability (similar to the CD-ROM). Though, it would not be as good for running the program due to there being not write or erase function once data has been written. However, it could be seen as overkill because the program will not use much of the data available on the DVD.

**USB-Drive**

USB Drives are very common portable forms of storage media. The biggest size currently available is 1TB. However, they come in various sizes, so to be more cost efficient the user could get one as low as 512MB. The USB Drive can read files at a rate of 92MB/s and write at 70MB/s (USB 3.0). This is considerably faster than the CD and DVD drives. The other obvious advantage of this is that you can re-write and erase data so this would be useful for updating the database when using the program.

USB drives now come in sizes as small as 2.2cm. This means that they could be easily misplaced, which may result in sensitive data being misused by other people. However, this issue can be combatted by using methods of encryption so unauthorized access doesn’t take place.

USBs are useful as a backup of the program or to use the program at home as they tend not to be very reliable and they can fail often. Also due to its size as mentioned previously, it is probably not the safest device to run the entire program on a regular basis.

**Hard Disk Drive**

Hard Disk Drives are the main method of secondary storage on a PC due to their size, reliability and read/write speeds. They currently range from 4GB to 8TB of storage capacity. This will be more than adequate to store the program on. However, the hard drive will most likely have something like an OS so it will also have to be able to store this. There are also two types of HDD’s: external and internal. They operate exactly as they say; the external is a HDD outside of the case of the computer as is connected using a USB, while internal HDD’s are connected using SATA cables and are inside the case.

Security of internal hard drives are very good because they are inside the case and someone would have to either take the whole computer or take off the case cover and unplug it and take it out of its slot to steal it.

The read/write speed is measured in rotations per minute. The current typical sort of speed is 7200rpm and the disk to buffer rate is a maximum of just over 1GB/s.

Internal HDD drives would be fantastic for running the program due to its speed and reliability. It would also have no problem storing it. Also the church would not have to invest in a new storage device as they already have one of these in their computer with enough storage capacity to store the program.

**Cloud Storage**

Files on a cloud storage based platform require an internet connection to be accessed. These are a very good method of backing up the program due to it being able to be accessed from anywhere with an internet connection and also the servers tend to be reliable if you use a platform like Google Drive to store the data. However, it cannot be easily updated by a program so it would not be a good program to run the program off it.

**Recommendation**

The recommendation to distribute the software is a DVD-ROM as it will be a way to send the user the program with no sensitive data on it. It will also be big enough to store the data on and it could also stay as a back-up of a clean install of the program and database. The cloud based storage would be the best form of back-up as you can now get 15GB of free storage so it is a very cheap way of storing data (if you have an internet connection). It is also very safe as companies like Google and Microsoft invest lots of money in security of data so they will have state of the art encryption to avoid any of the user’s sensitive data getting out. As for overall running of the program, it is recommended the users should use an Internal Hard Disk Drive as it is a very reliable way of running the software and it is not easily stolen due to it being inside the computer case. It will also meet the storage and read/write requirements of the program.

2.10 Identification of algorithms using structured English or pseudo code

**Rota Generation**

The rota needs to be automatically generated for the user. This means we need to take as many variables as we can and then get the program to create a rota suited to the users’ needs. This algorithm is designed to do this. It will go through each “cell” in the table and then assign it a volunteer. It will then go through the volunteers by ID and then see if they can be in the cell. If they can then they will insert into the cell, however if they don’t meet the requirements then the next volunteers will be put through the algorithm. Once all of the cells have been accounted for, the algorithm will finish.

For R ← 2 to MaxJobsAvailable

For C ← 2 to LastSundayInMonth

Loop until x ← MaxVolunteers or JobFilled ← True

If Volunteer[x]. TimesON LESS THAN MaxTimesOn

Loop until y ← Volunteer[x]. NoOfJobsQualified

If volunteer[x]. JobsQualified[y] ← Rota(R,1)

VolunteerCanDoJob ← True

End

End

Y=1

If VolunteerCanDoJob ←True

If VolunteerTotalDaysUnavailable NOT 0

Loop until y ← volunteer[x]. TotalDaysUnavailable

If Volunteer[x]. Unavailability[y] NOT Rota (1, C)

Rota (R, C) ← Volunteer[x]. Firstname ““ Volunteer[x]. LastName

Volunteer[x]. TimesOn← Volunteer[x]. TimesOn +1

End

End

End

End

End

End

End

**Display Rota**

This algorithm will insert each of the volunteers which have been pre-determined by the Rota Generation algorithm beforehand. It will simply have 2 loops (with one nested inside the other) and then it will go through the “cells” inserting the data into them.

For R ← 2 to MaxJobsAvailable

For C ← 2 To LastSundayInMonth

Rota (R, C). TextBox ← Rota (R, C)

End

End

**Insertion Sort**

This will sort any of the data that you wish by using an algorithm similar to this. This specific algorithm will sort the volunteer ID’s in numerical ascending order. It uses two loops to do this. With z being a holder and x and y being a counter in a form loop.

For x ← 1 to MaxVolunteers

z ← Volunteer(x).ID

For y ← x - 1 To 0 Step -1

If Volunteer(j).ID LESS THAN OR EQUAL TO z, Then Exit For

Volunteer (y + 1).ID ← Volunteer(y).ID

Next y

Volunteer (y + 1).ID ← z

Next x

**Binary Search**

This is an algorithm that is recursively designed. This means it is a procedure that recalls itself. It first of all will find the midpoint of the data in the database. It will then use this data and determine if the data you are searching for is higher or lower than the midpoint. If it is higher then it will take one off the midpoint and then set that as the MaxVolunteerID. However, if it is higher then it will do the opposite. If the data is found then it will return true, however if it is not then it will return false.

BinarySearch

If MaxVolunteerID LESS THAN MinVolunteerID,

Return Keyfound as false

OR

Mid ← midpoint [MinVolunteerID, MaxVolunteerID]

If Volunteer[mid].ID MORE THAN [Input]

MaxVolunteerID ← mid -1

Recall BinarySearch

OR If Volunteer[mid].ID LESS THAN [Input]

MaxVolunteerID ← mid +1

Recall BinarySearch

OR

KeyID ← mid

Return KeyFound ← True

End

End

**2.11 Description of needed queries using SQL**

**Login**

The data that the user has inputted into a text box will need to be compared to the data in the database. They will first need to compare the username to ensure that the user exists. We will then need to see if the password matches the one with the username.

SELECT Login(Username)

FROM Login

WHERE Login.Username ← [input]

SELECT Login(Password)

FROM Login

WHERE Login.Password ← [input]

**Rota**

When viewing the rota, it will just need the rota ID to find out where the rota is.

SELECT Rota(ID)

FROM Rota

When creating a new rota. All of this data will be needed to create the rota

SELECT Volunteers (Firstname, Lastname, JobsQualified, ID), Unavailability(Date)

FROM Volunteers

AND Unavailability

Then once the rota has been completed. To create statistics, you will need to load these into the database for the volunteer statistics. The statistics will be generated by the program and will require no user input.

SELECT vStatistics (NoTimesOn, NoOfTimesDoneJob, NoOfDaysUnavailable)

FROM vStatistics

WHERE vStatistics (NoTimesOn, NoOfTimesDoneJob, NoOfDaysUnavailable) = [Output]

SELECT cStatistics (AverageJobsQualifiedFor, NumberOfNewPeoplePerYear)

FROM vStatistics

WHERE cStatistics (AverageJobsQualifiedFor, NumberOfNewPeoplePerYear) = [output]

**Volunteer**

This is where the volunteers will be added. This will modify, add or delete a volunteer by taking the users input and then adding it to the selected database

SELECT Volunteers (Firstname, Lastname, JobsQualified, ID, Email, MaxTimesOn, Birth)

FROM Volunteer

WHERE Volunteers (Firstname, Lastname, JobsQualified, ID, Email, MaxTimesOn, Birth) = [Input]

**Jobs**

This is the same as volunteers, but for jobs

SELECT Job (Name, Description, ID)

FROM Job

WHERE Job (Name, Description, ID) = [Input]

**Unavailability**

This is the same as volunteers, but for unavailability

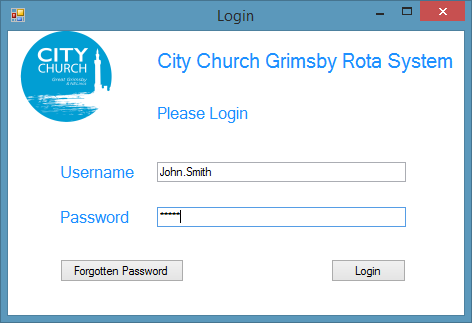
SELECT Volunteer (Firstname, Lastname, ID)

From Volunteer

WHERE Volunteer (Firstname, Lastname, ID) = [Input]

**2.12 Screen Designs**

**Logging** In



2

1

3

4

The first screen that the user will be greeted with is the login screen. This is where they will login using their user name and password. If they have forgotten their password, then they can change it. The colour scheme will be based around the colours in the logo (top left). It contains 4 Labels, 2 text boxes, 2 buttons and 1 image.

This is where the username will be entered. An example data of “John. Smith” has been used to show the sort of format that the username should be in. The text in this textbox will be validated or not after the user has pressed the login button.

1

The user will enter their password here. The password will be a pre-defined. The characters will be replaced in a similar format to the password above. This will also be validated when the user presses the log in button.

2

This button that will take the user to the main screen. However, it will only take the user to the main screen if the Username and Password match the data that is stored on the system. If it is not correct it will display an error message and the user will remain on the same screen.

3

This button will be useful if for any reason that the user has forgotten his/her password as they can change their password here. This button will simply take them to a screen where they can begin to change their password.

4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Username\_  TextBox | Text Box | String | John. Smith | N/A |
| 2 | Password\_  Textbox | Text Box | String (Data Hidden) | Password123 | N/A |
| 3 | Login\_Button | Button | Mouse Click | N/A | frmMainScreen |
| 4 | ForgottenPassword\_Button | Button | Mouse Click | N/A | frmSelectUserForgottenPassword |

Forgotten Password



3

2

1

This is the first screen that the user will see after they have pressed ‘ForgottenPassword\_Button’, it will take them to a screen where you will enter their security question. This is so the program will know what security question to load and what password to change.



7

5

6

4

This is the next screen after the user presses next. This will be to verify that it is the correct user that is trying to change their password. They will have to answer a pre-determined security question.



11

9

10

8

This screen is where the user will change their password. All the user has to do is put the same password in twice and then it will change the password to that. After this the user can login.

This is where the user must enter a valid username into a textbox. The username will be checked if it is valid when the user presses next. The same example data is used as in the previous user name box.

1

1

This button will check the data in the username text box and, if it is correct, it will go onto the next screen. If it is not, the user will see an error message and be asked to re-enter the username.

2

3

This button will take the user back to the login screen.

This is a read-only text box that will show the user a pre-determined question which they will have to answer to get onto the next stage in changing their password.

4

The user will input the answer to the question into this box. This will then be validated after they press the ‘change password’ button.

6

5

This will check the data in the Answer\_textbox and see if it is correct and then, if it is it, will move onto the next scree. If not, then the program will display an error message and the user will remain on the current screen.

7

This button will take the user back to the previous screen.

8

This text box will contain the password the user wishes to have.

This will contain the password the user wishes to have again for validation purposes. This password will be check with the one inputted into the NewPassword\_Textbox.

9

This will check the two text boxes and then will change the password and update the database. It will then take the user back to the login screen where they can then login.

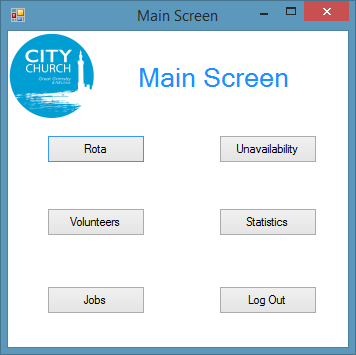
10

11

This button will do the same as number 7.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Username\_  Textbox | Text Box | String | John.Smith | N/A |
| 3 | Next\_button | Button | Mouse Click | N/A | FrmforgottenPassword |
| 3 | Back\_Button | Button | Mouse Click | N/A | frmLoginScreen |
| 4 | SecurityQuestion\_Textbox | Text Box | String | How old is City Church? | N/A |
| 5 | Answer\_  TextBox | Text Box | String | 5 years | N/A |
| 6 | Next\_button | Button | Mouse Click | N/A | frmForgottenPassword |
| 7 | Back\_Button | Button | Mouse Click | N/A | frmLoginScreen |
| 8 | NewPassword\_Textbox | Text Box | String (Data Hidden) | Password123 | N/A |
| 9 | ConfirmPassword\_Textbox | Text Box | String (Data Hidden) | Password123 | N/A |
| 10 | Change\_  Button | Button | Mouse Click | N/A | frmLoginScreen |
| 11 | Back\_Button | Button | Mouse Click | N/A | frmSelectUserForgottenPassword |

Main Screen



2

4

5

6

3

1

This screen is made entirely of buttons. It is the main hub of the program. The user can access everything from here.

1

This is a button that will take the user to the rota screen.

2

This is a button that will take the user to the volunteers’ screen.

This is a button that will take the user to the jobs screen.

3

4

This is a button that will take the user to the unavailability screen.

5

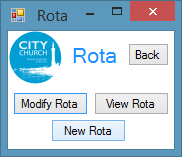
This is a button that will take the user to the statistics screen.

6

This button will log the user out and take the user back to the user screen.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Rota\_Button | Button | Mouse Click | N/A | frmRota |
| 2 | Volunteers\_  Button | Button | Mouse Click | N/A | frmVolunteers |
| 3 | Jobs\_Button | Button | Mouse Click | N/A | frmJobs |
| 4 | Unavailability\_Button | Button | Mouse Click | N/A | frmUnavailability |
| 5 | Statistics\_  Button | Button | Mouse Click | N/A | frmStatisticsSelection |
| 6 | LogOut\_  Button | Button | Mouse Click | N/A | frmLoginScreen |

Rota

On this screen the user will have the option to modify a rota, delete a rota or create a new rota. They can also have the option to go back to the main screen. This is the main hub for the rota part of the program.

1

4

3

2

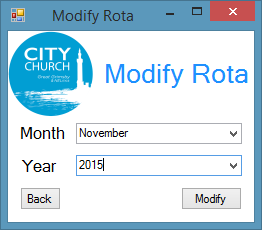
1

This button will take the user back to the main screen.  
  
This button will take the user to the Modify Rota screen.  
  
This button will take the user to the Delete Rota screen.  
  
This button will take the user to the New Rota screen.

2

3

4



5

8

7

6

5

The user will use the drop-down box to select the month of the rota they wish to edit.

6

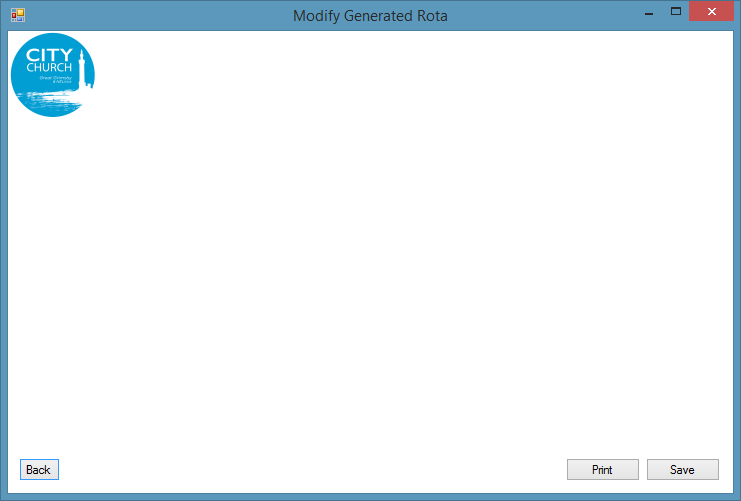
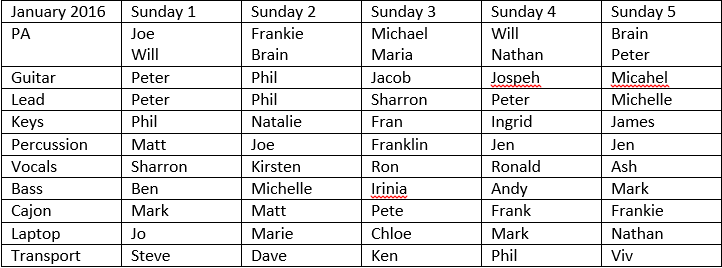
The user will use this drop-down box to select the year of the rota they wish to edit

7

This is a button that will take the user back to the previous screen.

This will load up the rota using the data in the two drop down boxes and then will load up the correct rota if it is found and will take the user to a screen displaying the rota. If there is no rota found, then the user will be displayed an error message and they will remain on the current screen.

8



12

11

10

9

This is an example of a displayed rota that can be modified using dropdown boxes. When the user has finished editing it, they will then have the option of printing, which will go into a print wizard, and then save. This will overwrite the existing file. Or the user can press back which will take them back to the previous screen without saving any changes.

This is example data of the rota. The client can change the rota using the template that will work better for each rota. There will be a different rota layout, but the concept will be the same.

9

This button will take the client back to the previous screen without saving any changes. Before they can do this they will be prompted whether or not they wish to go back and their changes will not be saved. If they press yes, then they will be taken to the previous screen. If they press no, the screen they were on will still be displayed.

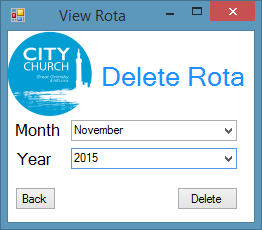
10

This button will load up a print wizard and will print off the rota in a printer friendly format. This means that the user can print the rota off easily. When this is done they will return to the screen they were on when they pressed the ‘print’ button.

11

This button saves the rota. This will simply overwrite the previous rota. However, the user will be asked if they wish to save just in case they pressed it by accident.

12

Here the user can delete a selected rota. After they have deleted a rota it will remove the file from their system. It will only work if there is a rota been made in the selected month and year.

13

14

15

16

13

The user will use the drop-down box to select the month of the rota they wish to edit.

14

The user will use this drop-down box to select the year of the rota they wish to edit.

15

This is a button that will take the user back to the previous screen.

16

This will use the data from the two drop-down boxes and it will try to find the corresponding rota and then delete it from the computer. If it cannot be find, it then will display an error message and not do anything.



This is the screen that the user can create a new rota. They just simply enter the month and the year of the rota they wish to create and the program will generate a rota using all of the variables it is given.

17

18

19

20

17

The user will use the drop-down box to select the month of the rota they wish to create.

18

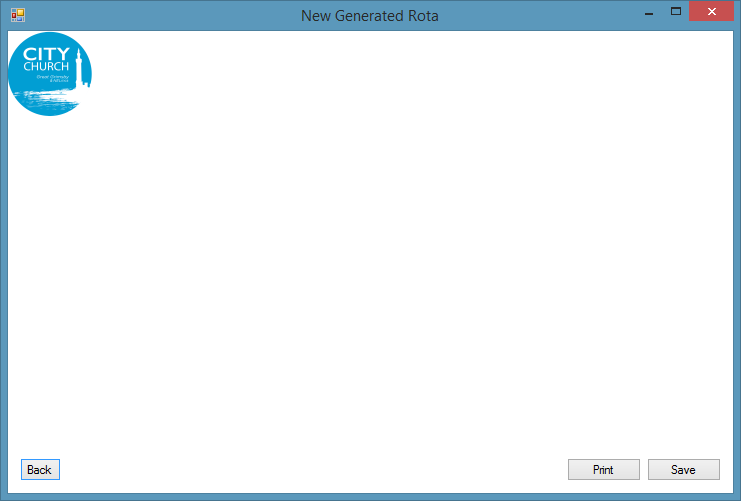
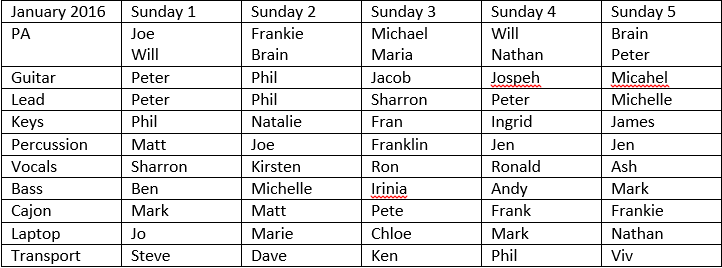
The user will use this drop-down box to select the year of the rota they wish to create.

19

This is a button that will take the user back to the previous screen.

This will generate a new rota using all the variables that it is given. It will use all of the databases available. It will both input and output to databases. After the rota has been created, it will take you to the next screen which will display the generated rota.

20



24

23

22

21

This screen will display the user’s newly generated rota. It is also editable using dropdown boxes so if there are any modifications that the user need to make, they can make these changes.

This is an example of a generated rota. The user can edit this rota using dropdown boxes that will display available volunteers.

21

22

This will take the user back to the previous screen without saving any changes. However, they will be prompted by the program asking if they really want to go back.

This button will load up a print wizard and will print off the rota in a printer friendly format. This means that the user can print the rota off easily. When this is done they will return to the screen they were on when they pressed the ‘print’ button.

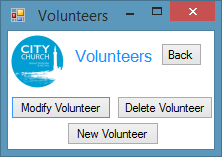
23

This will save the rota in the user’s files. The file name will be selected by the program based on the month and year that the rota is for. It will then save it and the user will return to the generated rota.

24

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Back\_Button | Button | Mouse Click | N/A | frmRota |
| 2 | ModifyRota\_  Button | Button | Mouse Click | N/A | frmModifyRota |
| 3 | DeleteRota\_  Button | Button | Mouse Click | N/A | frmDeleteRota |
| 4 | NewRota\_  Button | Button | Mouse Click | N/A | frmNewRota |
| 5 | Month\_Drop | Combo Box | Any of combo items | November | N/A |
| 6 | Year\_Drop | Combo Box | Any of combo items | 2015 | N/A |
| 7 | Back\_Button | Button | Mouse Click | N/A | frmRota |
| 8 | Modify\_  Button | Button | Mouse Click | N/A | frmModifyRotaGenerated |
| 9 | GeneratedRota\_Table | Table | Any of combo items | *See picture* | N/A |
| 10 | Back\_Button | Button | Mouse Click | N/A | frmModifyRota |
| 11 | Print\_Button | Button | Mouse Click | N/A | *Print Wizard* |
| 12 | Save\_Button | Button | Mouse Click | N/A | *Save As Wizard* |
| 13 | Month\_Drop | Combo Box | Any of combo items | November | N/A |
| 14 | Year\_Drop | Combo Box | Any of combo items | 2015 | N/A |
| 15 | Back\_Button | Button | Mouse Click | N/A | frmRota |
| 16 | Delete\_Button | Button | Mouse Click | N/A | *Confirmation Box* |
| 17 | Month\_Drop | Combo Box | Any of combo items | November | N/A |
| 18 | Year\_Drop | Combo Box | Any of combo items | 2015 | N/A |
| 19 | Back\_Button | Button | Mouse Click | N/A | frmRota |
| 20 | Generate\_  Button | Button | Mouse Click | N/A | frmNewRotaGenerated |
| 21 | GeneratedRota\_Table | Table | Any of combo items | *See picture* | N/A |
| 22 | Back\_Button | Button | Mouse Click | N/A | frmModifyRota |
| 23 | Print\_Button | Button | Mouse Click | N/A | *Print Wizard* |
| 24 | Save\_Button | Button | Mouse Click | N/A | *Save As Wizard* |

Volunteers

On this screen the user will have the option to modify a volunteer, delete a volunteer or create a new volunteer. They can also have the option to go back to the main screen. This is the main hub for the volunteer part of the program. 

4

3

2

1

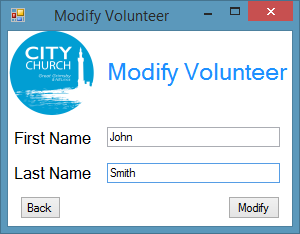
1

This button will take the user back to the main screen.  
  
This button will take the user to the Modify Volunteer screen.  
  
This button will take the user to the Delete Volunteer screen.  
  
This button will take the to the New Rota screen.

2

3

4

This screen will allow the user to select a volunteer to modify. Text boxes are used instead of combo boxes because there will be around 50 volunteers and will be too many to scroll through a combo box. The program will then check if the volunteer exists and then allow the user to modify the volunteer.

5

6

7

8

5

This is a text box where the user will enter the volunteer they wish to modify their first name.

6

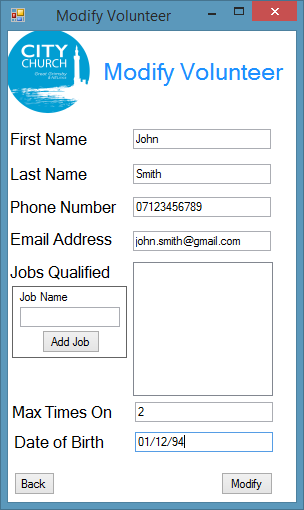
This is a text box where the user will enter the volunteer they wish to modify their last name.

7

This button will return the user to the previous screen.

This button will compare the user’s data to the data in the database and try and find a match. If it does, the user will go onto the next screen. However, if it doesn’t, the user will not be allowed to go onto the next screen and will get an error message. They will stay on the same screen.

8

This is the page that will load data from the database and allow the user to modify all of it. Once this is done the user will just press modify and the changes are saved immediately.

9

10

11

12

13

14

15

16

17

18

19

9

In this text box the user can modify the selected user’s first name.

9

10

In this text box the user can modify the selected user’s last name.

11

In this text box the user can modify the user’s phone number.

12

In this text box the user can modify the user’s email address.

13

This is a read-only list box that will show the jobs the volunteers are qualified for.

14

This text box is used to add jobs to the list of jobs the volunteers are qualified for.

This button will update the jobs the volunteers are qualified for by taking the data from the job name text box (13) and then putting it into the database and putting it in the jobs qualified textbox (13).

16

15

This is the text box where the user can modify the maximum times a volunteer can be on in one month.

17

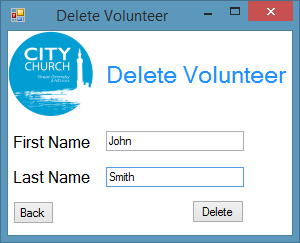
The user can update the volunteers DOB. This is only an option box though.

18

This button will take the user to the previous screen.

This is the button that will take all of the data in the text boxes and upload them to the database and save the changes. After this is done it takes the user back to frmVolunteers.

19

In this screen the user you can delete a volunteer. They just need to have the text boxes as an actual volunteer in the database and the volunteer will be removed.

21

22

23

20

20

The user will enter the volunteer’s first name in this text box.

22

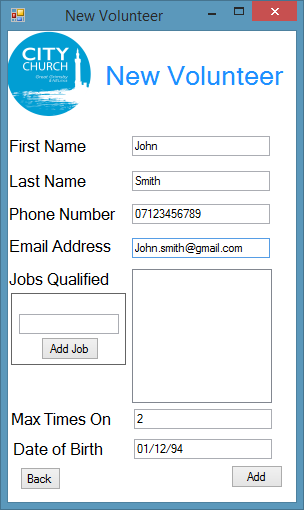
21

23

The user enters the volunteer’s last name in this text box.

This button will take the user back to the previous screen.

This will check the volunteer database with the data inputted into the text boxes and it will delete the chosen volunteer. If there is no volunteer that the user has inputted in the database, then they will be prompted with an error message and will be remain on the same screen.

This screen will allow the user to add a new volunteer to the database. They will have to have all of the right data in the right places and the volunteer will be added to the volunteer database.

24

30

32

31

29

28

27

26

25

33

34

24

In this text box the user can add the selected user’s first name.

25

In this text box the user can add the selected user’s last name.

26

In this text box the user can add the user’s phone number.

27

In this text box the user can add the user’s email address.

28

This is a read-only list box that will show the jobs the volunteers are qualified for.

29

This text box is used to add jobs to the list of jobs the volunteers are qualified for.

This button will update the jobs the volunteers are qualified for by taking the data from the job name text box (13) and then putting it into the database and putting it in jobs qualified textbox (13)

31

30

This is the text box where the user will add the maximum times a user can be on in one month.

32

The user can add the volunteers DOB. This is only an option box though.

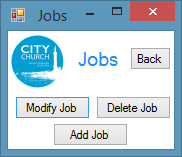
33

This button will take the user to the previous screen.

Once this button has been pressed, it will upload all of the data to the volunteer’s database and will then return the user to the volunteer’s screen.

34

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Back\_Button | Button | Mouse Click | N/A | frmRota |
| 2 | ModifyVolunteer\_Button | Button | Mouse Click | N/A | frmModifyVolunteerSelection |
| 3 | DeleteVolunteer\_Button | Button | Mouse Click | N/A | frmDeleteVolunteer |
| 4 | NewVolunteer\_Button | Button | Mouse Click | N/A | frmNewVolunteer |
| 5 | FirstName\_  TextBox | Text Box | String (letters only) | John | N/A |
| 6 | LastName\_  TextBox | Text Box | String (letters only) | Smith | N/A |
| 7 | Back\_Button | Button | Mouse Click | N/A | frmVolunteers |
| 8 | Modify\_  Button | Button | Mouse Click | N/A | frmModifyVolunteer |
| 9 | FirstName\_  TextBox | Text Box | String (Only Letters) | John | N/A |
| 10 | LastName\_  TextBox | Text Box | String (Only Letters) | Smith | N/A |
| 11 | PhoneNumber\_Textbox | Text Box | Integer (11 digits exactly) | 07123456789 | N/A |
| 12 | EmailAddress\_TextBox | Text Box | String (email address format) | [John.smith@gmail.com](mailto:John.smith@gmail.com) | N/A |
| 13 | JobQualified\_  ListBox | List Box | String (Letters only) | 4 | N/A |
| 14 | JobName\_Textbox | Text Box | String (Only letters) | PA | N/A |
| 15 | AddJob\_  Button | Button | Mouse click | N/A | *Adds job to Volunteer Database* |
| 16 | MaxTimesOn\_Textbox | Text Box | Integer (below 5) | 2 | N/A |
| 17 | DateOfBirth\_  Textbox | Text Box | String (Date format) | 01/12/94 | N/A |
| 18 | Back\_Button | Button | Mouse click | N/A | frmModifyVolunteerSelection |
| 19 | Modify\_  Button | Button | Mouse click | N/A | *Updates Volunteer Database* |
| 20 | FirstName\_  TextBox | Text Box | String (Only letters) | John | N/A |
| 21 | LastName\_TextBox | Text Box | String (Only letters) | Smith | N/A |
| 22 | Back\_Button | Button | Mouse click | N/A | frmVolunteers |
| 23 | Delete\_Button | Button | Mouse click | N/A | *Deletes volunteer* |
| 24 | FirstName\_TextBox | Text Box | String (Only Letters) | John | N/A |
| 25 | LastName\_  TextBox | Text Box | String (Only Letters) | Smith | N/A |
| 26 | PhoneNumber\_Textbox | Text Box | Integer (11 digits exactly) | 07123456789 | N/A |
| 27 | Email Textbox | Text Box | String (email address format) | [John.smith@gmail.com](mailto:John.smith@gmail.com) | N/A |
| 28 | JobQualified\_  ListBox | List Box | String (Letters only) | 4 | N/A |
| 29 | JobsQual\_Textbox | Text Box | String (Only letters) | PA | N/A |
| 30 | AddJob\_Box | Button | Mouse click | N/A | *Adds job to Volunteer Database* |
| 31 | MaxTimesOn\_Textbox | Text Box | Integer (below 5) | 2 | N/A |
| 32 | DOB\_Textbox | Text Box | String (Date format) | 01/12/94 | N/A |
| 33 | Back\_Button | Button | Mouse click | N/A | frmModifyVolunteerSelection |
| 34 | Add\_Button | Button | Mouse click | N/A | *Updates Volunteer Database* |

Jobs

On this screen the user will have the option to modify a job, delete a job or create a new job. You can also have the option to go back to the main screen. This is the main hub for the jobs part of the program.

1

2

3

4

1

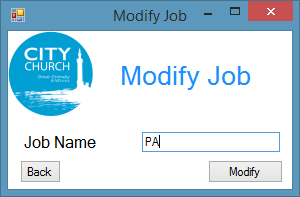
1

This button will take the user back to the main screen.  
  
This button will take the user to the Modify Job screen.  
  
This button will take the user to the Delete Job screen.  
  
This button will take the user to the New Job screen.

2

3

4

This is where the user will be able to select the job they wish to edit. They will type in the job name and can modify it if the job they have entered exists.

5

6

7

5

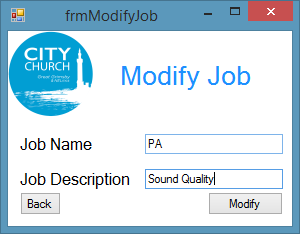
This is a text box where the user will enter the name of the job they wish to edit. There are too many jobs to use a combo box so a text box is used and a search can be carried out on the jobs database.

6

This will take the user back to the previous screen

This will search the database for the data in the textbox and then if it finds it, it will take the user onto the next screen. If not, then an error message will be displayed and the user will remain on the current screen.

7



This will allow the user to modify the data of the selected job. It will load up the data from the database and fill the text boxes so the user can modify the data. Once the user is done he/she will press modify and it will update the jobs database.

11

8

9

10

8

This textbox will display the job name data from the database ready for the user to edit.

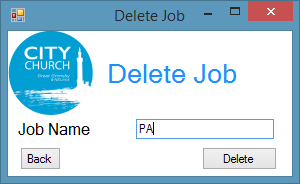
9

This textbox will display the job description data from the database ready for the user to edit.  
This will take the user back to the previous screen.

11

10

This will update the database with the data in the two textboxes.

 The user will delete a job from the data from this screen. They will enter the job they wish to delete into the textbox and press delete and, if the data in the text box matches any of the jobs in the database, then it will delete all data associated to that item.

12

14

13

12

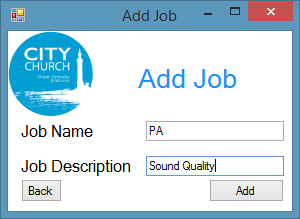
The user will enter the job name they wish to delete here.

13

This will take the user to the previous screen.

14

When the user presses this button, it will check the data from the textbox and compare it to the data in the database and then it will delete the job and all associated data with it. However, if it doesn’t then it will display an error message and the user will remain on the same screen.

This will allow the user to add a job into the job database. They just need to fill all of the text boxes in correctly and then press add and then it will be added to the database.

18

17

16

15

15

This is where the user will add the name of the job. This should be an easy to remember name.

16

This is the job description. It should be short, but tell the user the nature of the job.

17

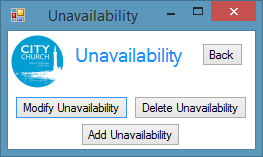
This will take the user to the previous screen.

This will update the jobs database with all of the data on the text box. Once it is pressed, it will add the job to the jobs database.

18

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Back\_Button | Button | Mouse Click | N/A | frmMainScreen |
| 2 | ModifyJobs\_  Button | Button | Mouse Click | N/A | frmModifyJobSelection |
| 3 | DeleteJob\_  Button | Button | Mouse Click | N/A | frmDeleteJob |
| 4 | AddJob\_  Button | Button | Mouse Click | N/A | frmDeleteJob |
| 5 | JobName\_  Textbox | Textbox | String (letters only) | PA | N/A |
| 6 | Back\_Button | Button | Mouse Click | N/A | frmJobs |
| 7 | Modify\_  Button | Button | Mouse Click | N/A | frmModifyJob |
| 8 | JobName\_Textbox | Textbox | String (letters only) | PA | N/A |
| 9 | JobDesc\_  TextBox | Textbox | String (letters only) | Sound Quality | N/A |
| 10 | Back\_Button | Button | Mouse Click | N/A | frmModifyJobSelection |
| 11 | Modify\_  Button | Button | Mouse Click | N/A | *Updates Jobs Database* |
| 12 | JobName\_  Textbox | Textbox | String (letters only) | PA | N/A |
| 13 | Back\_Button | Button | Mouse Click | N/A | frmJobs |
| 14 | Delete\_Button | Button | Mouse Click | N/A | *Updates Jobs Database* |
| 15 | FirstName\_  TextBox | Textbox | String (letters only) | PA | N/A |
| 16 | LastName\_  TextBox | Textbox | String (letters only) | Sound Quality | N/A |
| 17 | Back\_Button | Button | Mouse Click | N/A | frmJobs |
| 18 | Add\_Button | Button | Mouse Click | N/A | *Updates Jobs Database* |

Unavailability

On this screen the user will have the option to modify unavailability, delete unavailability or add unavailability. You can also have the option to go back to the main screen. This is the main hub for the unavailability part of the program.

1

2

3

4

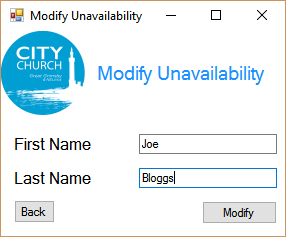
1

This button will take the user back to the main screen.  
  
This button will take the user to the Modify unavailability screen.  
  
This button will take the user to the Delete unavailability screen.  
  
This button will take the user to the add unavailability screen.

2

3

4



When you wish to modify someone’s unavailability you will be greeted by this screen. This will ask for the volunteers first and last name so that it can identify the volunteer and load up their unavailability.

5

6

7

8

5

The volunteer’s first name goes here. This will be used to find the volunteers unavailability.

6

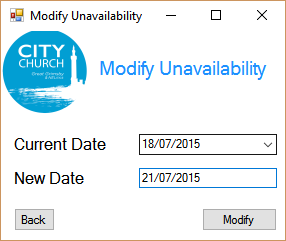
The volunteer’s last name goes here. This will be used to find the volunteers unavailability.

7

This will take the user back to the previous screen.

8

This button will look at the data in the text boxes and check it again the data in the database. It will then load their unavailability

 This will load up the selected volunteer’s unavailability and display it in a combo box. You will then be able to add a new date in the next textbox and then press modify and it will then update the unavailability.

9

10

11

12

This will display all of the current unavailability dates of the volunteer. You will then select the date you wish to modify.

10

9

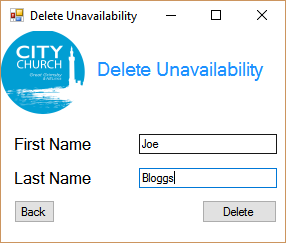
You will enter the date that you wish to change it to in this box

This will take you back to the previous screen

11

This will update the unavailability in the database. It will replace the one selected with the one in the text box

12



You will select the volunteer that you wish to delete one of their unavailability. You just need to have their first and last name and you will be able to do this. Once you press delete it will search the data base for the volunteer and if it finds any it will take you onto the next screen.

13

14

16

15

You will enter the first name of the volunteer that you wish to remove one of their unavailability.

13

You will enter the last name of the volunteer that you wish to remove one of their unavailability.

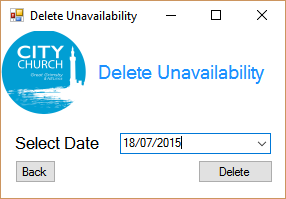
15

14

This will take you to the previous screen

This button will look at the data in the text boxes and check it again the data in the database. It will then load their unavailability

16

 This is the screen where you will select the date you wish to delete using the combo box. The user will then press delete and it will delete the selected unavailability from the database.

17

19

18

This will contain all of the unavailable dates of the unavailability. The user will use this date to select the date.

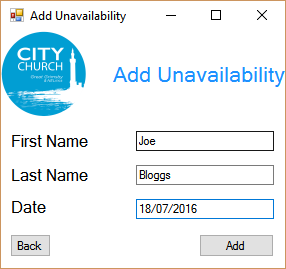
18

17

This will take you back to the previous screen.

19

This will delete the selected unavailability from the unavailability database.

 This is the screen where you will add unavailability. You just need to enter all of the required fields and press add. This will then update the unavailability database and add the unavailability.

20

21

22

24

23

20

This is where you add the first name of the volunteer you wish to add a date of unavailability

21

This is where you add the last name of the volunteer you wish to add a date of unavailability

22

This is where you add the date that the volunteer is unavailable

23

This will take you to the previous screen

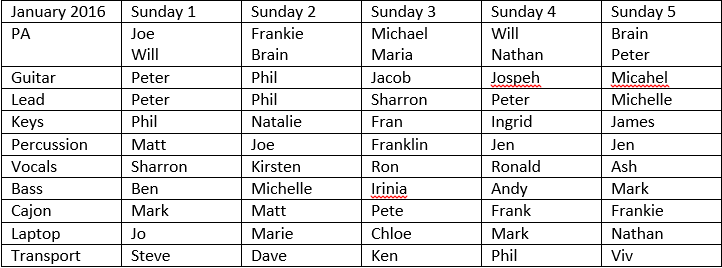
24

This will add all of the data from the text boxes to the unavailability database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Accepted Data | Example Data | Destination |
| 1 | Back\_Button | Button | Mouse Click | N/A | frmMainScreen |
| 2 | ModifyUnavailability\_Button | Button | Mouse Click | N/A | frmModifyUnavailability |
| 3 | DeleteUnavailability\_Button | Button | Mouse Click | N/A | frmDeleteUnavailability |
| 4 | AddUnavailability\_Button | Button | Mouse Click | N/A | frmAddUnavailability |
| 5 | FirstName\_TextBox | Textbox | String (letters only) | John | N/A |
| 6 | LastName\_Textbox | Textbox | String (letters only) | Smith | N/A |
| 7 | Back\_Button | Button | Mouse Click | N/A | frmUnavailability |
| 8 | Modify\_Button | Button | Mouse Click | N/A | frmModifyUnavailabilityDates |
| 9 | CurrentDate\_Drop | Combo Box | Selected items | 11/11/15 | N/A |
| 10 | NewDate\_Textbox | Textbox | String (Date format) | 11/11/15 | N/A |
| 11 | Back\_Button | Button | Mouse Click | N/A | frmModifyUnavailability |
| 12 | Modify\_Button | Button | Mouse Click | N/A | *Updates Unavailability Database* |
| 13 | FirstName\_TextBox | Textbox | String (letters only) | John | N/A |
| 14 | LastName\_Textbox | Textbox | String (letters only) | Smith | N/A |
| 15 | Back\_Button | Button | Mouse Click | N/A | frmUnavailability |
| 16 | Modify\_Button | Button | Mouse Click | N/A | frmDeleteUnavailability |
| 17 | Date Drop | Combo Box | Selected items | 11/11/15 | N/A |
| 18 | Back\_Button | Button | Mouse Click | N/A | frmModifyUnavailability |
| 19 | Delete\_Button | Button | Mouse Click | N/A | *Updates Unavailability Database* |
| 20 | FirstName\_TextBox | Textbox | String (letters only) | John | N/A |
| 21 | LastName\_Textbox | Textbox | String (letters only) | Smith | N/A |
| 22 | Date\_Textbox | Textbox | String (Date format) | 11/11/15 | N/A |
| 23 | Back\_Button | Button | Mouse Click | N/A | frmModifyUnavailability |
| 24 | Add\_Button | Button | Mouse Click | N/A | *Updates Unavailability Database* |

Outputs

This is everything that the program will output either in a PDF or similar format or print.

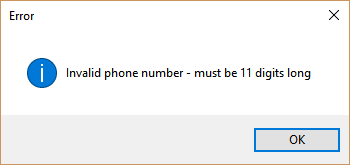
This is so that the user can easily access certain item without having to go into the program or in this case to distribute it to all of the volunteers without having to give them access to the program as they don’t need access to the program. This below is what the user can print and put into PDF format. It is a rota with a template that me and the client have agreed on as the rota they currently are using will not work with the system that I am creating.

This will then be distributed to all of the volunteers using the program, so they can view it using their device. The user can also choose to give people hard copies as I know some people in the church don’t have an email. This rota will vary with each user however the concept will be the same. The only thing that will be changed is the date depending on when they are creating it, if there are 4 or 5 Sundays in the month, the name of the jobs and obviously the volunteers in the boxes as currently these are all made up names.

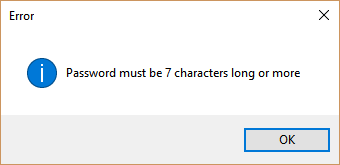
Error Messages

There will be many varying messages throughout the program, however most of them will follow a template so I will condense the amount of error messages down as much as I can.

**Length not correct**

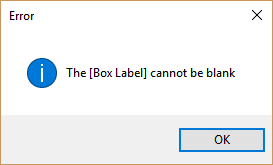
The first type of length check will be on the phone number as all phone number must be 11 digits long in order to be valid. If it is not equal to 11 digits, then this error message will be displayed:

The next length check will be done when the user is entering a new password as it must be 7 digits long. This is the error message box that will be displayed when a password is not the correct length:

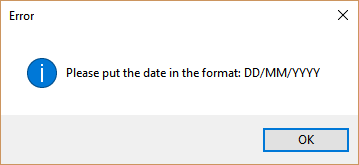


**Validation Checks**

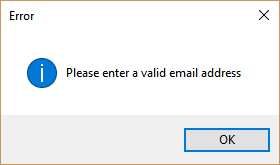
There will be many other validations check on screen, I will list all of them and then put a screen shot next to them:



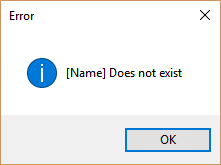
This is when a text box is left blank. It will recognize that a box is blank and then tell the user what box it is.



If the date is not in the correct format as shown in the error box, then it will flag the user up on it and then ask them to put it in the correct format



If the input in the email address does not contain a @ symbol, then it will be flagged up to the user telling them that the email address that they have sent is not valid.



If what the user has inputted does not exist in the database, then the user will be told that it doesn’t exist in this format.

2.13 Security and Integrity of Data

Security of Data

The definition of security of data is preventing un wanted people from accessing data, so that we can follow the data protection act.

Username and Password

Each user will have their own individual password so that each of them will have access to the correct volunteers and they won’t have access to people that they don’t need to. This will avoid any of them using the data they have not inputted and don’t have permission to view. Also there will be checks to make sure that the correct username matches the correct password to avoid any of the users being able to login to any account. Passwords will also be case sensitive.

Access to Sensitive Data

To ensure that no data is leaked. I will be separating the databases. This is so that people cannot view data that they have permission to access. So even if two users have the same user in their database it should detect the same information and then add them to the list of volunteers that they can access. However, if they don’t add them to their list then they will not be able to see this volunteer.

Recommendations for The User

I would recommend that the user keeps the program inside an operating system that has a password on it. I would also recommend that this be on an account that they only have access to, to avoid the volunteers’ data being accessed by unwanted personnel.

System being recovered

If the user ever faces the fact that the data in their system has been corrupt or lost, I would recommend that they recover it using the latest data that they have backed up. You would simply, copy the files from the backup onto the secondary storage that you wish to use. If the user wishes to use the same storage that they had before, they must make sure that the storage device has been fixed before using it again.

Due to there being the possibility that the data could become corrupt, I would make regular backups on a secure storage device to ensure that in the instance that anything does go wrong you will not have lost too much important data.

Integrity of Data

Data integrity is ensuring the accuracy and consistency of data that is stored in a database. This will ensure that none of the data become corrupt.

Length Checks

Many Length check will be taken place in the program to ensure all of the data is correct. The biggest and most important length check that will take place is the check to ensure that the passwords are long enough. I have decided that a password of 7 characters should be good enough for this program. Users will also be encouraged to use capital letters and maybe special characters. We will also make sure that people phone number is 11 characters long. An example of this would be this:

If Len[Input] ← 7

Add [Input] to the database

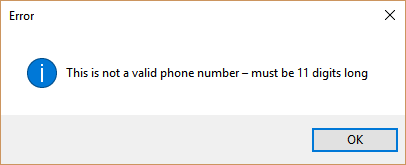
Else

Output [“This is not a valid phone number – must be 11 digits long”]

End

There would also be a similar check to ensure that the passwords are long enough.

Message box example:



Blank Input Check

There will be checks on every single piece of inputted information to ensure that the box isn’t left blank. The only exception to this would be the Date of Birth box in the volunteers’ database as this is an optional bit of data. An example of this wold be

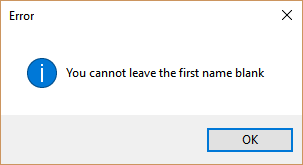
If [Input] ← “”

Output [The [box label] cannot be blank]

Else

[Add to database]

End



The error message would be similar to this:

Duplicate Information

Certain bits of information cannot be entered twice. An example of this is a volunteers with the same first name, last name and phone number. This is so that the program is as efficient as possible. Due to the size of the database it will take little time to search the database to see if a user is trying to create a volunteer again or anything else similar to this. The algorithm would look like this.

For 1 to MaxVolunteers

If [Input] ← DBFirstName

If [Input] ← DBLastName

If[Input] ← DBPhoneNumber

Output [“This user already exists”]

End

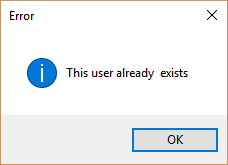
End

End

Else

Add user to database

End

The error message would look like this: