



3002ICT/7701ICT - INDUSTRY PROJECT

GROUP 8 - CHIMERA SOFTWARE

GCHHS C.H.I.L.D.S. Team Referral Database

PROJECT PLAN

Project Manager:

Jarrold CABALZAR

Team Members:

Adam BOIKO

Michael LERRO

Matt LARTER

Supervisor:

Steven TUCKER

Date Submitted: October 13, 2013

Due Date: October 13, 2013

Contents

1	Project Organisation	4
1.1	Organisation Structure	4
1.2	Organisational Boundaries and Interfaces	5
1.3	Project Responsibilities	7
1.3.1	Project Member Roles	8
1.4	Project Charter	8
2	Project Scope	10
2.1	Project Justification	10
2.2	Software Development Life Cycle Model	11
2.3	Scope Statement	13
2.4	Scope Verification	14
2.4.1	Product Deliverables	14
2.5	Scope Control	16
2.6	Project Requirements	18
2.6.1	Core Requirements	18
2.6.2	Non-Core Requirements	18
3	System Design and Overview	19
3.1	Application Overview	19
3.2	Assumptions, Dependencies and Constraints	20
3.2.1	Assumptions	20
3.2.2	Dependencies	20
3.2.3	Constraints	21
3.3	Monitoring and Controlling Mechanism	21
3.3.1	Redmine	21
3.3.2	Client Meetings	22
3.3.3	Project Team and Supervisor Meetings	22
3.4	Software Requirement Specification	22
3.4.1	Core Requirements	23
3.4.2	Non-Core Requirements	23
3.5	System Architecture	23
3.6	Integrated System Structure	24

4	User Interface Design	26
4.1	Component Hierarchy Chart	26
4.2	Design Prototypes	27
5	Project Management Planning	31
5.1	Staff Scheduling	31
5.1.1	GANTT Chart	32
5.1.2	PERT Chart	33
5.2	Work Breakdown Structure	34
5.3	Risk Management	35
5.3.1	Risk Responses	37
5.4	Staff Responsibilities	38
5.5	Communication Management	39
5.6	Resource Requirements and Allocations	40
6	Technical Processes	42
6.1	Methods, Tools and Techniques	42
6.1.1	Methods	42
6.1.2	Tools	43
6.1.3	Techniques	44
6.2	Work Product Documentation	45
6.3	Project Support Functions	45
6.3.1	Quality	45
6.3.2	Verification/Validation	46
7	Cost Analysis	47
7.1	Cost Estimation	47
7.2	Cost Control	49
8	Testing Plan	51
8.1	Test Plan Objectives	51
8.2	Testing Strategies	52
8.3	Test Cases	53
8.4	Test Schedule	55
8.5	Appendix	56

1 Project Organisation

This section of the document will go into detail on the methods with which the project stakeholders will communicate with each other and their respective roles and responsibilities.

1.1 Organisation Structure

The project manager, Jarrod Cabalzar, will act as the primary spokesperson and overseer for the group. The team consists of Michael Lerro, Matt Larter, and Adam Boiko who will split between them the duties of Documentation, Software Development and Testing. The specific duties of the team are outlined in the diagram below.



Figure 1: Project organisational structure.

1.2 Organisational Boundaries and Interfaces

This section elaborates on how the communication between people involved in the project will communicate.

Team Contact

Communication via a Facebook group is the primary method of inter-group communication. It provides an easy, convenient, and quick method of contacting all group members simultaneously. Alternate methods of communication include emails, phone calls, text messages, and Redmine comments.

Supervisor Contact

Weekly half-hour meetings with Steven, the project's supervisor, will be conducted with the entire group present to ensure that all group members are on task and performing as expected. This is also a chance for the group to get advice and discuss any possible problems with the project.

Client Contact

The group will endeavor to contact the client, Lisa Hutchinson, at least once a week via email or other means in order to provide her with the opportunity to voice any project-related concerns. This is also to ensure we are on the right track with regards to development, and that progress is being achieved at a sufficient rate. Face-to-face meetings are conducted as needed, as regular meetings aren't considered necessary by the group.

The following table provides an idea of the identified communication methods and scheduled physical meetings for our group with each other and with the client and project supervisor.

Communication Type	Objectives	Medium	Frequency, Priority	Audience	Owner	Deliverable
Email	Communication	Email	Frequently, Low	Stakeholders, Project team	Project Manager	Agenda, meeting minutes
Facebook	Communication	Web	Frequently, Low	Project team	Project Manager	
Initial Meeting	Get acquainted with client, gather requirements	Face-to-face	Once, Very High	Stakeholders	Project Manager	
Specific Meeting	Technical issues, change requests	Face-to-face	As required, High	Stakeholders, Project team	Project Manager	
SVN and Redmine	Used to submit and record progress	Web	Daily, Medium	Project team	Project Manager	
Weekly Meeting	Assign/review tasks and discuss design and implementation	Face-to-face	Weekly, Medium	Project team	Project Manager	Status and progress reports, meeting notes

Table 1: Communication strategies

1.3 Project Responsibilities

This section provides a listing of project roles and the members which are assigned those roles. It will go into mild detail of what such roles entail, and thus each persons' overall goal in the project.

Project Manager

The project manager's main objective is to oversee the project and to keep it within budget and time constraints. In order to do this, the project manager will be required to: approve major changes to the project, assign issues to team members, handle any conflicts, organise any team meetings, maintain contact with the client, and various other high-level jobs.

Software Developer

The software developer's primary job in the project will be to design, implement and test the code for the application. This includes following a project schedule as laid out by the project manager, and following a chosen Software Development Life Cycle model. The developer will also be responsible for overseeing usability testing with the client.

Editor

The editors' primary job will be the creation of any documentation, formal invitations, product box art, and other such matters. Other side priorities include, but aren't limited to, the analysis of software requirements for the product, assessment of project risk, the responsibility to update the Documentation of the project following reviews by the Group Manager. The Editor is also responsible for developing a change management approach, with an associated Change Request Form, which when completed the editor will update the Project Requirements for the Project, which will be completed during the first Presentation.

1.3.1 Project Member Roles

Member	Role	Responsibilities	Contact
Adam Boiko	Team member	Editor	adam.boiko@griffithuni.edu.au
Jarrold Cabalzar	Project Manager	Software developer / Project manager	jarrod.cabalzar@griffithuni.edu.au
Matt Larter	Team member	Editor	matt.larter@griffithuni.edu.au
Michael Lerro	Team member	Software developer	michael.lynch-lerro@griffithuni.edu.au

Table 2: Project member roles & contact details

1.4 Project Charter

Stakeholders	Role
Lisa Hutchinson	Client
Adam Boiko	Team Member
Michael Lerro	Team Member
Jarrold Cabalzar	Team Member
Matt Larter	Team Member

Table 3: Project stakeholders & their respective roles

Client History

CHILDS is a collaboration of nurses who help children with early childhood illnesses by conducting checkups on children in clinics. CHILDS has received a grant from gr8 start to facilitate prevention, early intervention, early identification and intervention initiatives.

CHILDS conducts vision screening, hearing assessments, and developmental assessments. All data about the checkup was traditionally entered through the use of an Excel spreadsheet. There are, however, a number of problems with this approach: efficiency and distribution of data being the primary concern. In regards to efficiency many of the common tasks performed are quite manual, data would have to be manually merged into spreadsheets, and any statistics and graphs derived from that data would also have to be done manually. This is one of the primary advantages of the application our group is proposing - everything is automated. Data is synchronised with a master server, and graphs and statistics are

a click away.

Success criteria

1. The application has met the requirements specified by the client
2. The final budget was within the limits of the specified quote
3. The application passed usability and functionality testing and was found to be easy to use and thoughtfully laid out
4. The application contains no critical defects and can run and be used successfully
5. The user is able to successfully write and retrieve data from the master database
6. The security of the application prevents unauthorised access and inadvertent damage to the master database
7. The user is able to search for clients and generate reports based on the collected data
8. An administrator is able to promote existing users to administrator status
9. Regular users are not able to access totals or data relating to hubs other than their own
10. Query results can be exported as graphs or table in a variety of formats including .pdf and .jpg
11. If there is no existing network connection, the application stores any data modifications in the local database and then periodically queries the state of the connection until it is able to upload the data

2 Project Scope

This section of the project plan will justify the benefits of the project for client approval, the Software Life Cycle Model that will be in use and the justification of this decision. The scope will go into explicit detail about what the system must have and what it will not do, and a list of people who will require approval for the confirmation of the project and the processes they are responsible for, with a solid process changing the scope. The scope will also go into detail about the core and non-core requirements for the project.

2.1 Project Justification

The aim of the project is to provide Queensland Health nurses with a collection of data about client referrals and allow them to easily follow up on the outcomes of those referrals. It will also serve to justify the continual funding of the CHILDS team and their services.

The application will allow nurses to input data about a particular client's treatment, access client-specific records and view certain aggregate totals for their particular medical hub. Additionally, administrative users will be able to query the data, and produce tabular and graphical outputs, in order to help justify the continued funding and support of the team.

In essence, the application is a much-needed upgrade from the current spreadsheet-based system, which operates offline on individual data, to a centralised, concurrent system that can be securely hosted within the existing Queensland Health infrastructure.

2.2 Software Development Life Cycle Model

[Schwalbe, 2011].

The Agile Development model has been adopted for this project in order to maintain efficient development techniques and to help deliver a quality product that meets all specified requirements. By employing the Agile methodologies, the project team can quickly and efficiently adapt to changes in requirements whilst maintaining low costs and exposing software flaws in a timely fashion.

The communication methodologies defined in the Agile method will also be implemented, allowing the team to communicate effectively with all stakeholders involved in the project. Maintaining constant communication with stakeholders promotes feedback and encourages the client to become more involved with the development of the project which, in turn, will produce a higher quality product with less inconsistencies.

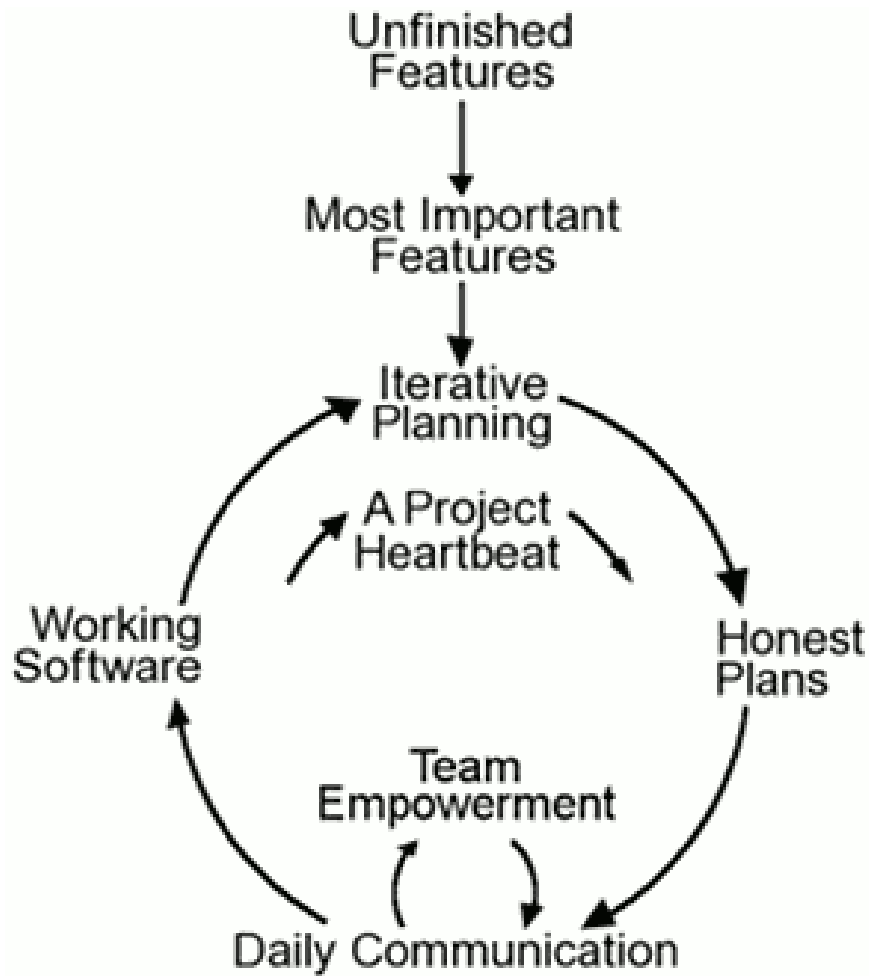


Figure 2: The Agile process

Figure 2 demonstrates the general process that encompasses the Agile approach.

The Agile Development Scrum Methodology will be employed for the duration of the project. Work items are derived from project requirements which are then broken down into tasks. These tasks are added to the appropriate sprint and then assigned to a member of the project team. Each sprint is a 2 week period of time in which the team completes several identified tasks, working towards a particular milestone within the development of the overall product. By using this method, the product can be reviewed at the end of each sprint, with modifications made

where necessary and defects within the software can be identified and rectified straight away. The use of the Scrum methodology also helps to keep the team motivated by working towards short-term goals rather than a single, finished product.

2.3 Scope Statement

Project Title: GCHHS C.H.I.L.D.S. Team Referral Database

Project Start Date: 25th of July 2013

Project Delivery and Finish Date: 25th of October 2013

After considering the requirements specified by the client during the initial meeting, it has been concluded that a desktop application would best meet these requirements. The proposed application will include facilities to input new client data, query the existing data and generate reports, in graphical or tabular format, which can then be exported for printing or presentations.

Although the collated data will be stored in a centralised database, the application is designed to be robust with regards to network connectivity, as nurses may not always have a stable Internet connection when attempting to input or query data. This will be achieved by having a local copy of the database with which new data can be added and existing data can be modified whilst offline. When a connection is established again, the user can then upload their changes to the master database thus allowing the application to withstand unreliable internet connections and other inherent network problems in a way that web-based applications cannot.

The application will be developed in C++ using the Qt framework. The user interface will primarily be tab-based. This design will help keep the functions and features of the software clean and easily accessible, and help prevent any visual clutter from an overabundance of options and buttons.

Connection to the central database will be established through an HTTPS connection which ensures that all confidential information will be guarded against any unauthorised access. Data stored locally will be encrypted and never transferred

or stored in memory in plain text.

Administrators will have unrestricted access to all data, while clinicians can only access data pertaining to their hub. Administrator accounts can only be created via an existing administrator.

2.4 Scope Verification

The Scope Verification process will be performed through inspections which are scheduled for weeks 6 and 12. This process includes a review of any deliverables, and the formal acceptance and sign-off by the client.

The list of deliverables to be signed off for both weeks 6 and 12 can be found in Section 2.8 of this document. There are three possible outcomes to the client's inspection of the deliverables:

- Accepted deliverable
- Unaccepted deliverable - request a change (week 6 only)
- Project document updates

2.4.1 Product Deliverables

Project Management:

1. Project Charter
2. Team Contract
3. Project Scope Statement
4. WBS
5. PERT chart
6. Critical Path
7. Risk Matrix
8. Schedule

9. Cost Baseline
10. Concept baseline
11. User requirements baseline
12. Analysis baseline
13. Design baseline
14. Construction baseline
15. Gantt Chart
16. Project Plan
17. Progress Reports
18. Final Project Presentation
19. Final Project Report
20. Meeting Notes

Product Deliverables:

1. Testing results
2. Training document
3. Change log
4. Acceptance agreement
5. Software packaging
6. User manual
7. Final source code package
8. Demonstration and Presentation of System

2.5 Scope Control

In the development of practically any project, changes are a given. Clients' change their minds, the priorities of the project may change, new features may be selected for inclusion, or any other of the multitude of possible events. The group has implemented a way to facilitate such changes in the change management procedure.

The first step in the change management procedure is the filling out of the Change Control Form, found on the next page of this document. Any person wishing to request a change should fill this out and present it to the project manager.

The project manager will then discuss the changes with the client, and if the change is agreed upon then the impact of the change will be evaluated by the project team.

The impact of the proposed change will include any potential changes to the budget, any changes to the project time-frame, any associated risk involved with the change, any change to the testing procedure, and any other possible considerations.

Finally, the change will be discussed with the client, project manager, the project team involved in facilitating the change, and the person responsible for requesting the change. This discussion will be the final decider if the change will indeed be implemented, and will be the final sign off on the change.

It should be noted that all changes should be approached in this manner. No oral agreements should be made, and no change should be implemented without the change request procedure being followed. This process is to prevent scope creep, keep the project in an agreed upon time-frame and budget, and to mitigate against being caught in a development-request spiral.

Change Control Form

Change Request #: _____ Project: _____

CHANGE REQUEST INITIATION: Originator: _____ Phone#: (____) _____ email: _____ Date Submitted: ____/____/____ System/Product/Service Name: _____ Version Number: _____				
CONFIGURATION ITEM: Software: ____ Firmware: ____ Hardware: ____ Documentation: ____ Other: _____				
CHANGE TYPE: New Requirement: ____ Requirement Change: ____ Design Change: ____ Other: _____				
REASON: Legal: ____ Market: ____ Performance: ____ Customer Request: ____ Defect: ____ Other: _____				
PRIORITY: Emergency: ____ Urgent: ____ Routine: ____ Date Required: ____/____/____				
CHANGE DESCRIPTION: (Detail functional and/or technical information. Use attachment if necessary.) Attachments: Yes / No				
TECHNICAL EVALUATION: (Use attachment to explain changes, impact on other entities, impact on performance etc.) Received By: _____ Date Received: ____/____/____ Assigned To: _____ Date Assigned: ____/____/____ Type of Software/Hardware/etc. Affected: _____ Modules/Screens/Tables/Files Affected: _____				
Documentation Affected: Requirements Specification System Design Specification System Test Plan Training Plan User System Reference Manual System Maintenance Manual Other (Specify)	Section # _____ _____ _____ _____ _____ _____	Page # _____ _____ _____ _____ _____ _____	Date Completed ____/____/____ ____/____/____ ____/____/____ ____/____/____ ____/____/____ ____/____/____	Initial _____ _____ _____ _____ _____ _____
TIME ESTIMATES to make the change: (Use attachment if necessary.)				
Lifecycle Stage Analysis/Design Coding/Testing Acceptance Total Hours:	Est. Time _____ _____ _____ _____	Act. Time _____ _____ _____ _____	Date Comp. ____/____/____ ____/____/____ ____/____/____	Remarks _____ _____ _____
Project Impact Analysis Needed: Yes / No (If yes, include impact on budget, resources, schedule, risk etc.)				
APPROVALS: Change Approved: ____ Change Not Approved: ____ Hold (Future Enhancement): ____ 1. Signature _____ Date: ____/____/____ 2. Signature _____ Date: ____/____/____ 3. Signature _____ Date: ____/____/____				

Reference: [Emprend Inc, 2006]

2.6 Project Requirements

This section will list the core and non-core requirements for the project, and their associated priorities as agreed upon by the project team and client.

2.6.1 Core Requirements

PR No.	Requirement	Priority
PR.01	Application allows for new Vision Screening, Hearing Assessment, Developmental Assessment, Health Promotion, and Meeting Attendance data to be input to the database	High
PR.02	Database connects to front-end	High
PR.03	Data can be imported to the database from .CSV file	Medium
PR.04	Data can be input into the database via UI for new records	High
PR.05	Data can be modified through UI for existing records	High
PR.06	Data can be backed up in .CSV format	Low
PR.07	Data can be displayed in Graphs and Tables	High
PR.08	User can query Database	Medium
PR.09	Data can be synced with the Master Database	Low
PR.10	Clinicians can access data pertaining to their hub only	Low
PR.11	Administrators have access to all data	Low
PR.12	Administrator account can only be created by an existing Admin.	Low

2.6.2 Non-Core Requirements

PR No.	Requirement	Priority
PR.01	Administrators can modify Clinician accounts.	High
PR. 02	Users can build their own queries within the application.	High

3 System Design and Overview

This section will outline core issues relating to the implementation of our project as a whole. Firstly, we will provide an overview of the application and discuss the project's assumptions, dependencies and constraints. Secondly, we will address the requirements of the software and how we will monitor and control the implementation of these features, and finally we will cover the architecture of the system and how we can integrate it into external systems.

3.1 Application Overview

"GCHHS C.H.I.L.D.S. Team Referral Database" is an application designed to provide convenient and accurate storage for confidential data for Queensland Health. This data is extracted through testing on school-aged children in various geographic clinics in Queensland. These tests range from vision, hearing and various developmental assessments. The application will also track referrals to doctors and specialists and will have search and refine capabilities. There will be various display methods within the application depending on the user that is logged in. Nurses and staff from a geographic centre will be able to view all records entered from their own centre and graph results, but not other geographic centres. Administrator accounts will be able to view records and graphed totals from any geographic centre as well as a universal total combining all geographic centres.

The application will run on client computers in the field, and will allow each person in the field to keep a synchronised and current copy of the database using a secure network connection. The application will use a local SQLite database where queries and modification can be made. The application can then either automatically or manually synchronise itself with the master database seamlessly and painlessly. The primary reason for choosing to use a local database is so that a continuous, uninterrupted connection to the network will not be required. Instead the data can be synchronised whenever a connection is available.

3.2 Assumptions, Dependencies and Constraints

During the course of development, assumptions have to be made about the knowledge potential users will have with regards to using our program. This section will detail some of those assumptions, as well as any dependencies and constraints the project team has made with regards to the software application.

3.2.1 Assumptions

Assumption	Details.
Language	User can speak and read English at a high standard.
Computer	User has access to a computer with Queensland Health network permissions.
Application	User is aware how to start the application.
Login	User has been issued with a CHILDS application login.
Qualifications	User is qualified to work with children and perform relevant assessments.
Manual	User has read the user manual and is aware how to enter test results correctly.

Table 4: Project Assumptions.

3.2.2 Dependencies

Dependency	Details
Queensland Health PC	User must be on the Queensland Health approved machine.
Network Access	Network must be functioning.
Login	User must have a valid login.
Child	User must have a child from their geographic centre to generate results.
Qualifications	User must be qualified to work with children and perform relevant assessments.

Table 5: Project Dependencies.

3.2.3 Constraints

Constraint	Details
Time	User must complete assessments within the time frame of the appointment/test.
Location	User cannot be on a machine that is not networked to Queensland Health.
Child	The child will likely require the company of a parent/school staff member.
Last Constraint	Insert details.

Table 6: Project Constraints.

3.3 Monitoring and Controlling Mechanism

It is integral that project team members have adequate monitoring and control of their goals and activities such that optimal performance, communication and efficiency can be reached.

Methods used to ensure adequate quality management of the project will include Redmine, regular client meetings, project team meetings, supervisor meetings and usability testing.

3.3.1 Redmine

The primary monitoring and control mechanism used for the duration of the project will be Redmine. This system incorporates several important features that will be integral to maintaining performance and efficiency during the completion of project tasks. Redmine provides the ability for users to create, edit, and assign individual tasks to group members. These are known as issues and can be grouped together to make up a milestone.

Redmine also provides the project team with a powerful file-sharing platform called subversion, which allows copies of documents and source code to be maintained and merged together by a group of developers smoothly and easily. It also allows all group members to retrieve any current or previous version of a file any time, anywhere.

3.3.2 Client Meetings

At least one meeting will take place with the project client - Lisa Hutchinson - in Week 2 of the project. Additional meetings may be scheduled between the project team and client in order to discuss important features and milestones. As the requirements will be solidified by Week 4, no client meetings will be needed to discuss software requirements.

3.3.3 Project Team and Supervisor Meetings

Regular meetings with the project team as a whole will take place with the project supervisor - Steven Tucker - at least once per week. This ensures adequate monitoring from the supervisor, and allows project team members to justify their work for the week. It also provides a regular opportunity for discussion on whatever project related issues may have arisen.

The project team will have less structure to its meeting schedule as discussions of different issues requires different meeting protocol. The group will mostly meet in part, as there is already a guaranteed meeting with the supervisor once per week.

3.4 Software Requirement Specification

The tables below currently list the functional and non-functional requirements for the Software Requirement Specification. The table provides a good overview on the products limits, but will improved and modified on future iterations of the project.

3.4.1 Core Requirements

PR No.	Requirement	Priority
PR.01	Application allows for new Vision Screening, Hearing Assessment, Developmental Assessment, Health Promotion, and Meeting Attendance data to be input to the database	High
PR.02	Database connects to front-end	High
PR.03	Data can be imported to the database from .CSV file	Medium
PR.04	Data can be input into the database via UI for new records	High
PR.05	Data can be modified through UI for existing records	High
PR.06	Data can be backed up in .CSV format	Low
PR.07	Data can be displayed in Graphs and Tables	High
PR.08	User can query Database	Medium
PR.09	Data can be synced with the Master Database	Low
PR.10	Clinicians can access data pertaining to their hub only	Low
PR.11	Administrators have access to all data	Low
PR.12	Administrator account can only be created by an existing Admin.	Low

3.4.2 Non-Core Requirements

PR No.	Requirement	Priority
PR.01	Administrators can modify Clinician accounts.	High
PR.02	Users can build their own queries within the application.	High

3.5 System Architecture

A person can either login to the application or register for an account. If the user requires administrative access, another administrator must elevate the user from the Configure Users option in the settings tab. The query tab allows users to pull outputs from the database based on generated queries, and generate statistics and graphs from them. The input tab allows records to be added to the database, and the settings tab allows the user to change various application settings.

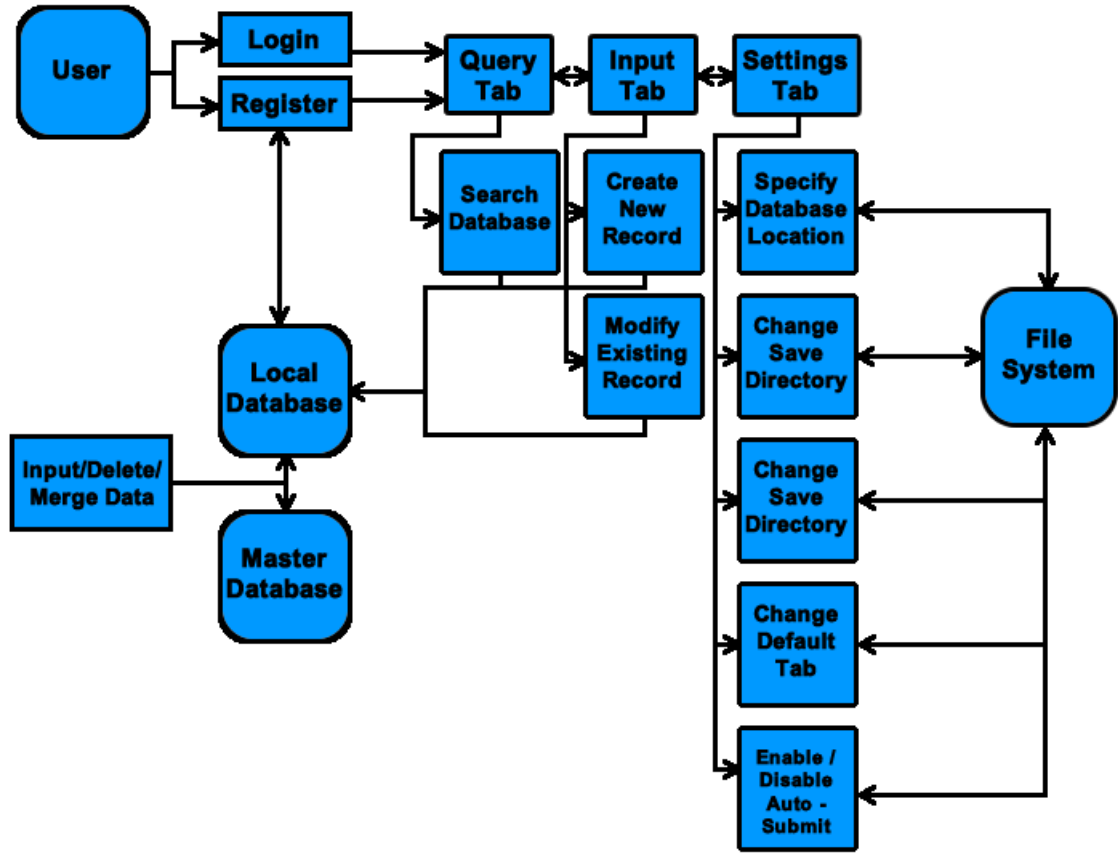


Figure 3: Use Case Diagram

3.6 Integrated System Structure

The program will be created using Qt Creator and written in C++ using the Qt framework, with SQLite being used as the local database back-end. The application will be centralised using a master server provided by the client. Our application will, however, use a local database such that a constant internet connection will not be required for continuous use of the program. When a network connection is established, the application will be able to synchronise the local database with master server, with no interruption to the user or their work.

Synchronising to the database involves merging any differences between the two databases, whether they be removals or additions.

The program can synchronise to the database either manually or automatically when a network connection is available. The database allows for a user to be associated with a regional hub, based on their local account preferences defined in our program settings.

4 User Interface Design

The application UI was designed to be easy to use and clear. A singular window encompasses all the primary functionality of the program, with a tabular layout providing clear differentiation between different user objectives. The user interface is, for the most part, basic. This is a design goal intended to accommodate our target audiences objectives.

4.1 Component Hierarchy Chart

The basic layout of our program is as shown below. The main window, as mentioned before, encompasses the entire program, providing a clear layout and making the objective of each program tab clear.

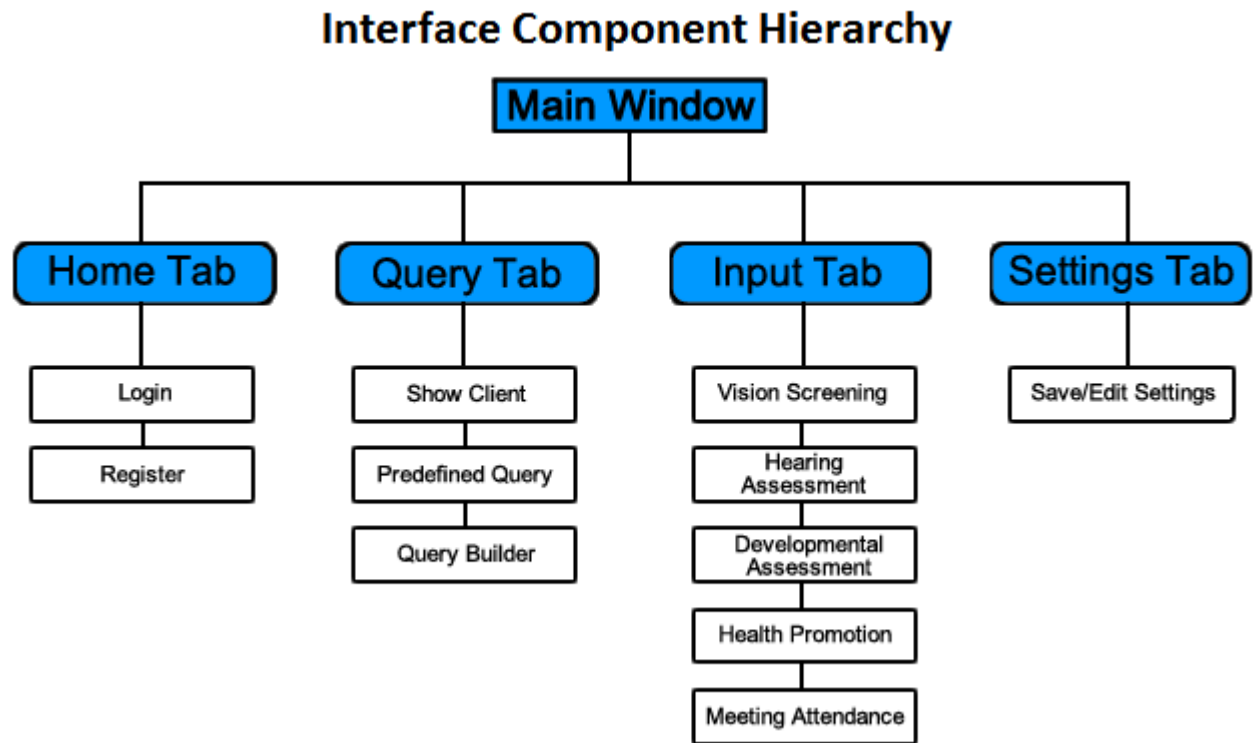


Figure 4: Program Component Hierarchy

4.2 Design Prototypes

Due to the nature of our program and the professional setting with which it will be used, the design team decided to keep bright colours and gratuitous animation and images to a minimum. We have opted for clean and clear sans-serif fonts, and a primarily black and white motif in our application user interface. Clarity, minimalism and ease of use are key, and obnoxiousness and novelty avoided. Some of the design prototypes for our application windows are provided below.

The UI is distributed across multiple tabs.

The Login and Register sections are mutually exclusive.

Users have the ability to select the hub that they work at when registering an account.

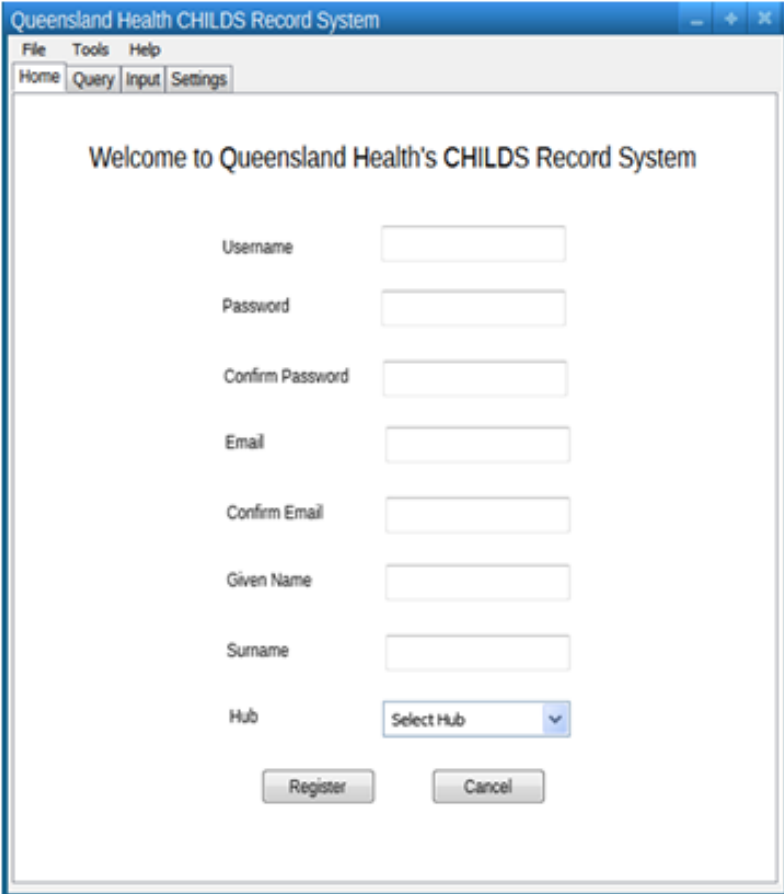


Figure 5: The Home tab

The home tab of the program. All other tabs are disabled until the user logs in. Should it be required, the user is also able to register for an account, upon which they can then log in.

Users can search for client-specific data, show a predefined query or build their own queries to generate reports.

Generated reports can be exported in both graph and tabular format to images or PDF files.

Queensland Health CHILDS Record System

File Tools Help

Home Query Input Settings

Show Patient 254865 Developmental Assessments GO

Show Complexity of Client Care GO

Year	Referrals
2010	365
2011	249
2012	651
2013	157

Graph Data

Figure 6: The Query tab

This tab allows statistical and graph outputs to be generated from the database.

Queensland Health CHILDS Record System

File Tools Help

Home Query Input Settings

Vision Screening Hearing Assessment Developmental Assessment Health Promotion Meeting At

Client ID

Date of Assessment

School/CCC Attended ▼

Screening Result ▼

Referred From CHILDS Team To ▼

Outcome of Referral ▼

Date Advice Received

Comments

Submit

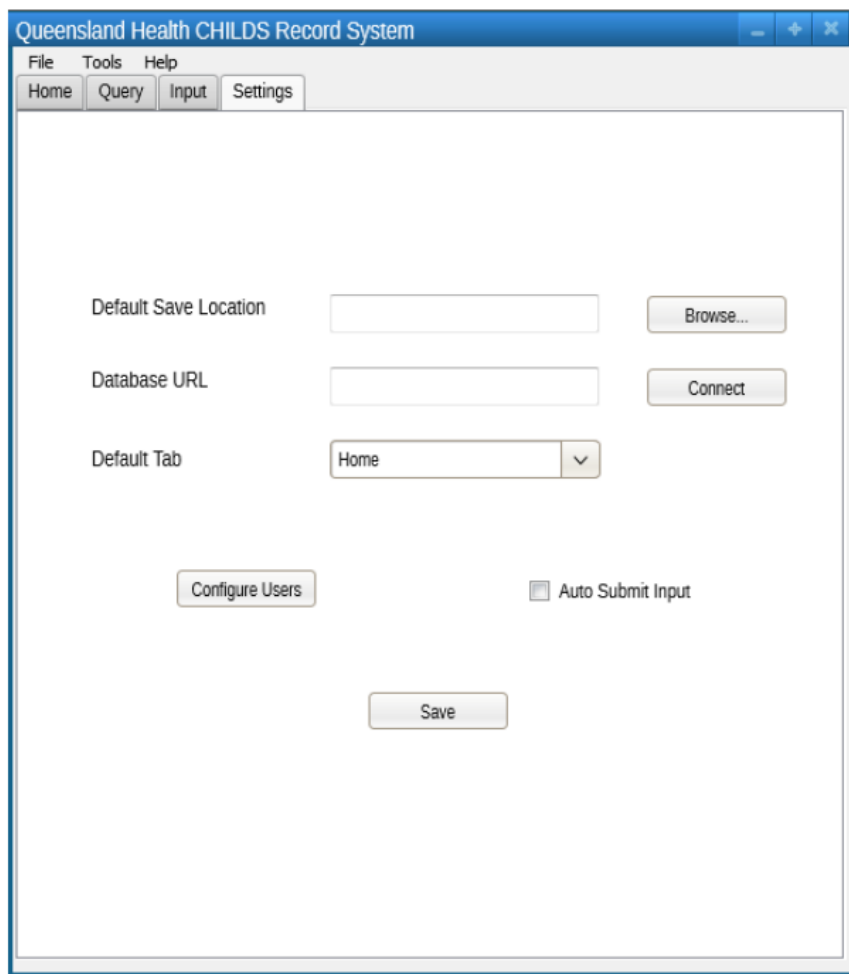
Auto-Submit Disabled

This tab allows data to be entered into the database.

Users can select to have the new data automatically submitted to the database when they change tabs.

Figure 7: The Input tab

This tab is where the user is able to input information into the database. Queries are generated dynamically based on the options the user selects - with a selection of common defaults provided.



The application settings available will change depending on whether the user is a standard user or admin.

Options were kept to a minimum - a primary objective in our application is to limit complexity and the potential for bugs. Adding too many settings will hinder those objectives

Figure 8: The Settings tab

As you can see, the primary considerations were minimalism, efficiency, and a thoughtful reduction of visual clutter. We intend the UI to be approachable to new users, and straight and to the point.

5 Project Management Planning

The project management planning section of the project explains key staff scheduling, Work Breakdown Structuring, Risk Management, Staff Responsibilities, Communication Management, and Resource Requirements and Allocations.

5.1 Staff Scheduling

The scheduling tools used by the project team are Microsoft Project and Redmine. The group will be able to manipulate the schedule by observing the baselines for the project, allowing us to determine any changes or actions required in updating the schedule. The team's schedule plan consists of a critical path methodology which tracks the estimated end of the project, giving an idea of the progress of the project team at any given time, and a rough estimate on when the project should be completed.

5.1.1 GANTT Chart

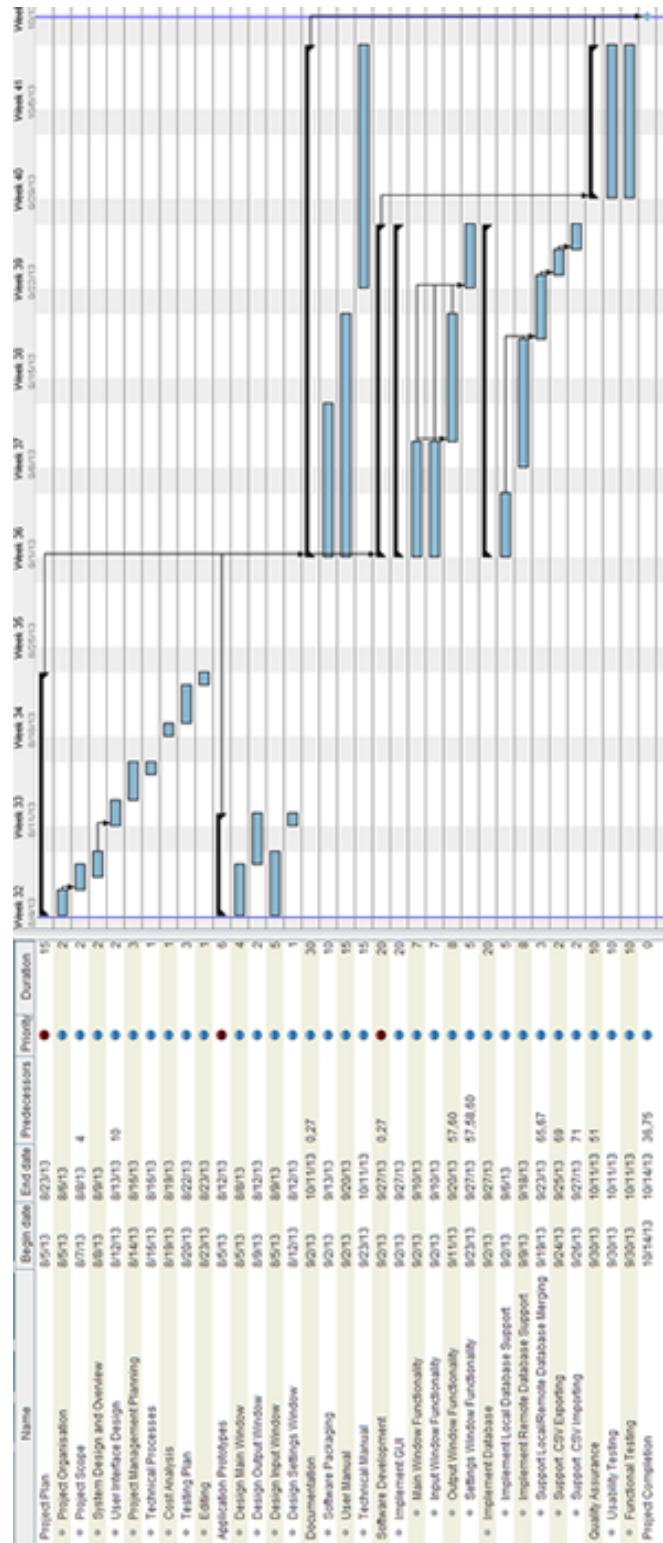


Figure 9: GANTT Chart

5.1.2 PERT Chart

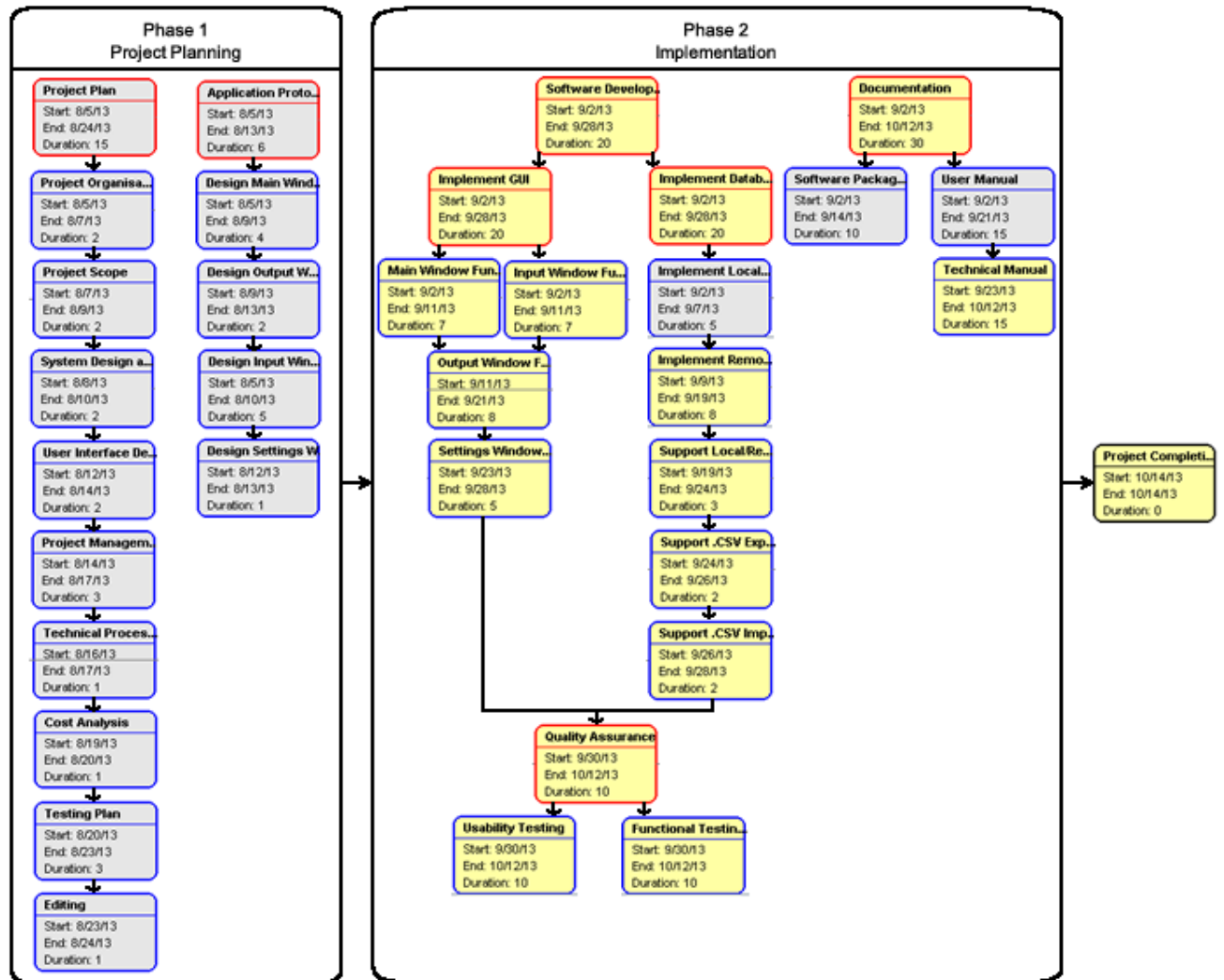


Figure 10: PERT Chart

5.2 Work Breakdown Structure

The table following this displays our WBS tree which shows the lowest level of tasks which is involved in the Work Breakdown Structure its parent tasks in display.

Name	Begin date	End date	Predecessors	Priority	Duration
Project Plan	8/5/13	8/23/13		High	15
Project Organisation	8/5/13	8/6/13		Normal	2
Project Scope	8/7/13	8/8/13	4	Normal	2
System Design and Overview	8/8/13	8/9/13		Normal	2
User Interface Design	8/12/13	8/13/13	10	Normal	2
Project Management Planning	8/14/13	8/16/13		Normal	3
Technical Processes	8/16/13	8/16/13		Normal	1
Cost Analysis	8/19/13	8/19/13		Normal	1
Testing Plan	8/20/13	8/22/13		Normal	3
Editing	8/23/13	8/23/13		Normal	1
Application Prototypes	8/5/13	8/12/13		High	6
Design Main Window	8/5/13	8/8/13		Normal	4
Design Output Window	8/9/13	8/12/13		Normal	2
Design Input Window	8/5/13	8/9/13		Normal	5
Design Settings Window	8/12/13	8/12/13		Normal	1
Documentation	9/2/13	10/11/13	0, 27	Normal	30
Software Packaging	9/2/13	9/13/13		Normal	10
User Manual	9/2/13	9/20/13		Normal	15
Technical Manual	9/23/13	10/11/13		Normal	15
Software Development	9/2/13	9/27/13	0, 27	Highest	20
Implement GUI	9/2/13	9/27/13		Normal	20
Main Window Functionality	9/2/13	9/10/13		Normal	7
Input Window Functionality	9/2/13	9/10/13		Normal	7
Output Window Functionality	9/11/13	9/20/13	57, 60	Normal	8
Settings Window Functionality	9/23/13	9/27/13	57, 58, 60	Normal	5
Implement Database	9/2/13	9/27/13		Normal	20
Implement Local Database Support	9/2/13	9/6/13		Normal	5
Implement Remote Database Support	9/9/13	9/18/13		Normal	8
Support Local/Remote Database Merging	9/19/13	9/23/13	65, 67	Normal	3
Support .CSV Exporting	9/24/13	9/25/13	69	Normal	2
Support .CSV Importing	9/26/13	9/27/13	71	Normal	2
Quality Assurance	9/30/13	10/11/13	51	Normal	10
Usability Testing	9/30/13	10/11/13		Normal	10
Functional Testing	9/30/13	10/11/13		Normal	10
Project Completion	10/14/13	10/13/13	36, 75	Normal	0

Figure 11: WBS Table for GCHHS C.H.I.L.D.S. Team Referral Database Project

Our WBS model is based on a Agile Scrum model SDLC. This was chosen due to its suitability to projects with a short time frame - just like ours.

5.3 Risk Management

The following table provides some potential risks that could possibly affect the outcome of the project, and displays any preventative measures used to prevent said risks.

ID	Risk/Event	Overall Impact	Preventative steps/response
E1	Skills lacking from Team members.	Occasional + Minor = 3D	With the use of online resources as well as work from previous years from similar works, to advance our respective skills to the necessary levels.
E2	Completion of functional requirements before schedule.	Seldom + Minor = 2D	The project will implement a large amount of testing therefore there will also be something to add.
E3	The final build contains many bugs.	Occasional + Critical = 3B	The project has a testing phase for each iteration of the build. This allows us to catch bugs as we progress and avoid a stock pile of bugs.
E4	The program does not connect to the internet.	Seldom + Negligible = 2E	We will ensure that the application can work decentralised, as well as centralised, so that if the program cannot access the internet from the associated clinic, then the data can be updated at a later date.

E5	Source Code is completely Lost.	Improbable + Negligible = 1E	Redmine allows the group to return to a previous version of a source code, therefore a complete loss cannot occur. Each team member has a locally stored version that can be updated from the repository at any time.
E6	Building phases for GUI/DB take longer than anticipated.	Likely + Moderate = 4C	The final two weeks of the project were originally for documentation completion, this time can also be for building phases for GUI and/or DB.
E7	Documentation is not fully completed.	Occasional + Critical = 3B	Team members are assigned specific parts of the documentation to complete, whilst the more difficult sections are completed as a group to ensure that the documentation is completed to the highest standard.
E8	Unable to update the applications database from hospital database.	Occasional + Critical = 3B	The applications final product can work decentralised whilst still allowing the database to update from a centralised database. Technical support could later repair such a problem.

Probability of Occurrences			Catastrophic	Critical	Moderate	Minor	Negligible
Definition	Meaning	Value	(A)	(B)	(C)	(D)	(E)
<i>Frequent</i>	<ul style="list-style-type: none"> Occurs frequently Will be continuously experienced unless action is taken to change events 	5	5A	5B	5C	5D	5E
<i>Likely</i>	<ul style="list-style-type: none"> Occur less frequently if process is corrected Issues identified with minimal audit activity Process performance failures evident to trained auditors or regulators 	4	4A	4B	4C	4D	4E
<i>Occasional</i>	<ul style="list-style-type: none"> Occurs sporadically Potential issues discovered during focused review. 	3	3A	3B	3C	3D	3E
<i>Seldom</i>	<ul style="list-style-type: none"> Unlikely to occur Minimal issue identification during focused review 	2	2A	2B	2C	2D	2E
<i>Improbable</i>	<ul style="list-style-type: none"> Highly unlikely to occur 	1	1A	1B	1C	1D	1E

Figure 12: Risk Matrix.

[PHE.com].

5.3.1 Risk Responses

Risk Avoidance, Risk Acceptance, Risk Transference, and Risk Mitigation, is used in the the project to define strategies for risk responses. The risk register, project management plan, and other project documents are all updated from the four features of the risk response.

Risk Acceptance Strategies

- Incorporate any risk management process into routine project processes, incorporate regular risk reviews, reports and updates.
- Document any of the decisions
- Develop a risk-aware culture for the project and organisation
- Take note of any identified risk and agree to actions in the project strategy
- Buffer for cost estimate, so that the budget will not go over or under budget

5.4 Staff Responsibilities

The group has been assigned project-length roles for the duration of this project, due to the GUI and Database implementation of the project requiring more attention by individual group members.

Jarrold Cabalzar acts as the Project Manager, and the primary spokesperson and overseer for the group. Other responsibilities include Software Development and Quality Assurance.

Adam Boiko will complete the Documentation, User and Technical Manuals, and Packaging.

Matt Larter will complete the Documentation, User and Technical Manuals, and Packaging.

Michael Lerro's responsibilities include Software Development and Quality Assurance.

5.5 Communication Management

Team Contact

Communication via a Facebook group is the primary method of inter-group communication. It provides an easy, convenient, and quick method of contacting all group members simultaneously. Alternate methods of communication include emails, phone calls, text messages, and Redmine comments.

Supervisor Contact

Weekly half-hour meetings with Steven, the project's supervisor, will be conducted with the entire group present to ensure that all group members are on task and performing as expected. This is also a chance for the group to get advice and discuss any possible problems with the project.

Client Contact

The group will endeavor to contact the client, Lisa Hutchinson, at least once a week via email or other means in order to keep her in the loop and ask queries with regards to the application. This is also to ensure we are on the right track with regards to development, and that progress is being achieved at a sufficient rate. Face-to-face meetings are conducted as needed, as regular meetings aren't considered necessary by the group.

5.6 Resource Requirements and Allocations

Resource Requirements

All of the resources for the project are provided by the group team members, including: human, hardware and software resources.

Human Resources

The project team will be completed by four team members:



Figure 13: Project team structure.

Software Resources

Resource	Description
Facebook	Team communication
Email	Informal client communication
MS Windows	Operating System
Redmine	Issue tracking software
Tortoise SVN	Subversion client
Latex	Create project plan
Pencil Project	Design Prototypes
Gantt Project	Gantt and Pert Diagrams
Adobe Reader	Project plan document
Adobe Fireworks	Create prototype interface images and other diagrams
Qt	Integrated Development Environment
SQLite	Database

Table 7: Software Resources

Learning Skills

All team members have the responsibility to research any new skills that will help accomplish any tasks to be undertaken during development. All tasks have an allocated timeslot for researching using web-based resources and tutorials, books and other media.

6 Technical Processes

The Technical Processes section details the Methods, Tools, Techniques that will be used to construct the software. The Work Product Documentation section states all deliverables for the project, which includes the product itself, and all associated documentation and contracts. Finally, the Project Support Functions section will clarify procedures to ensure that client quality requirements are met.

6.1 Methods, Tools and Techniques

This section identifies the various methods, tools and techniques used throughout the duration of the project.

6.1.1 Methods

Method	Description
Client Meetings	Meetings will be scheduled to display and discuss progress during the implementation stage. Usually at the request of the client.
Supervisor Meetings	30 minute meetings with the project supervisor once per week
Project Team Meetings	Informal meetings between project team members to discuss progress and to complete project work
Performance Reviews	Informal discussion between group members to evaluate performance and suggest any relevant improvements
Email Communication	Regular email communication between the project manager and the client enables swift attention to all concerns from both parties.

Table 8: Methods used and brief description

6.1.2 Tools

Tool	Description
Facebook	Team communication
Email	Informal client communication
MS Windows	Operating System
Redmine	Issue tracking software
Tortoise SVN	Subversion client
Latex	Create project plan
Pencil Project	Design Prototypes
Gantt Project	Gantt and Pert Diagrams
Adobe Reader	Project plan document
Adobe Fireworks	Create prototype interface images and other diagrams
Qt Creator	Integrated Development Environment
SQLite	Database
Valgrind	Memory checker

Table 9: Tools used and brief description

6.1.3 Techniques

Technique	Description
Usability Testing	Black-box usability testing will be conducted to ensure that the system is simplistic in appearance, yet fully-functional and robust under the covers.
Functionality Testing	White-box functionality testing will allow the system processes to be evaluated by iterating through predetermined input/output pairs and comparing the output with the expected result.
Change Request Form	A change request form is to be filed if it is determined that an item or task should be altered in any way other than the agreed upon state as of the time that the project plan was signed-off on.
Use Cases	Use Cases and their associated diagrams are established based on the requirements determined by the client and project team. They will be used to design the system processes.

Table 10: Techniques used and brief description

6.2 Work Product Documentation

This section covers all documentation that is produced by the project team and handed to the client throughout the duration of the project. These documents include a copy of the project plan, a contract stating acceptance of the project by the client and the project manager, a user manual for the end user, a copy of the product packaging and the final signed contract that will signal completion of the project.

Document	Description
Signed contract	The contract to be signed by the client and the project manager committing all parties to the development of the application.
Instructional manual	A hard copy and PDF copy (supplied with program) of how to use the program, and its available functions and features.
Product packaging	The product packaging.
Final signed contract	The final acceptance contract signed by the project manager and client.

Table 11: Product documentation and descriptions

6.3 Project Support Functions

This section covers the Project Support Functions that will be implemented by the project team to assure adequate Quality, Verification/Validation and Configuration of the product during planning and development.

6.3.1 Quality

The Quality requirements of the project are clearly displayed below. It is critical that these requirements are agreed upon by the project team and the client and strictly adhered to throughout the development of the product.

QR No.	Quality Requirement Description
1	User interface must maintain a consistent look and feel.
2	Users must be able to view records and summaries from their geographic centre only.
3	Administrators must be able to view records and summaries from all geographic centres.
4	Application must boot from any PC on the QH network and sync data to the most updated version.
5	Project team] must exchange weekly emails with the client to assess any quality concerns that either party may have.

Table 12: Project Quality Requirements.

6.3.2 Verification/Validation

The Project team will ensure Verification/Validation of Project deliverables via regular client meetings, usability testing and C++ code validation.

Client Meetings:

Client Meetings throughout the project will ensure that the client has the opportunity to verify and validate all concerns with progress and quality. These meetings can be called by the client or the project team provided that adequate notice is given.

Usability Testing:

Usability testing will be conducted throughout the project development which will ensure that the application look and feel is up to client standard. The specifics of this usability testing are discussed in Section 8.4 (Testing Plan).

C++ Code Validation:

The code written by the project team will undergo rigorous error-checking, specifically errors with memory corruption. The preferred software used will be Valgrind.

7 Cost Analysis

With regard to the cost of developing the GCHHS C.H.I.L.D.S. Team Referral Database application, human labour is by far the biggest consideration. Software costs are, in general, considered to be negligible to null due to the use of open source software. This section contains a breakdown of product costs, and how they will be managed.

7.1 Cost Estimation

Based on the following summaries and justifications, the estimated cost for the project has come to a total of **\$32,760**. This also includes a 20% safety margin in case something doesn't go according to plan. The cost breakdowns are as follows:

Project team cost breakdown

Role	Pay / hour	Hours / week	Total
Project manager	\$50	20	\$1000
Software developer	\$50	20	\$1000
Editor	\$50	20	\$1000

Task	Subtask	Duration	Project member	Labour cost	Other cost	Total cost estimate
Project Plan						
	Project Organisation	2 days	Editor	\$400	\$0	
	Project Scope	2 days	Editor	\$400	\$0	
	System Design And Overview	2 days	Editor	\$400	\$0	
	User Interface Design	2 days	Editor	\$400	\$0	
	Project Management Planning	3 days	Editor	\$600	\$0	
	Technical Processes	1 days	Editor	\$200	\$0	
	Cost Analysis	1 days	Editor	\$200	\$0	
	Testing Plan	3 days	Editor	\$600	\$0	
	Editing	1 days	Editor	\$200	\$0	
Cost subtotals				\$3400	\$0	\$3400
Application Prototypes						
	Design Main Window	4 days	Software developer	\$800	\$0	
	Design Output Window	2 days	Software developer	\$400	\$0	
	Design Input Window	5 days	Software developer	\$1000	\$0	
	Design Settings Window	1 days	Software developer	\$200	\$0	
Cost subtotals				\$2400	\$0	\$2400
Documentation						
	Software Packaging	10 days	Editors	\$2000	\$100	
	User Manual	15 days	Editors	\$3000	\$0	
	Technical Manual	15 days	Software developer	\$3000	\$0	
Cost subtotals				\$8000	\$100	\$8100

Software Development						
Implement GUI						
	Main Window Functionality	7 days	Software developer	\$1400	\$0	
	Input Window Functionality	7 days	Software developer	\$1400	\$0	
	Output Window Functionality	8 days	Software developer	\$1600	\$0	
	Settings Window Functionality	5 days	Software developer	\$1000	\$0	
Implement Database						
	Implement Local Database Support	5 days	Software developer	\$1000	\$0	
	Implement Remote Database Support	8 days	Software developer	\$1600	\$0	
	Support Local/Remote Database Merging	3 days	Software developer	\$600	\$0	
	Support .CSV Exporting	2 days	Software developer	\$400	\$0	
	Support .CSV Importing	2 days	Software developer	\$400	\$0	
Cost subtotals				\$9400	\$0	\$9400
Quality Assurance						
	Usability Testing	10 days	Editors	\$2000	\$0	
	Functional Testing	10 days	Software developer	\$2000	\$0	
Cost subtotals				\$4000	\$0	\$4000
20% Safety Margin						\$5460
TOTAL COST:						\$32760

With an additional **\$5460** - 20% of the total cost - added as a buffer, for a grand total cost of **\$32,760**.

7.2 Cost Control

Due to the lack of any real costs aside from the human factor, there is nothing much else to consider with regard to unexpected costs. The obvious exception to this is if the project blows its time constraints, and the developers are required to work longer than anticipated.

At a high level we intend to prevent such an occurrence - and to mitigate it should it actually occur - by using frequent and consistent communication with the client, and keeping every group member aware of the overall progress of the project - not just their own little corner. Also, if need be, lower priority features

can be sacrificed in favour of finishing the project earlier, neglecting the need for an increased budget - so long as the client is willing to forgo the elected features.

The project group has come up with some practical methods of dealing with cost underestimation, however. These include: a 20% overestimation for the total cost of the project, and a 4-step cost control and mitigation method:

- The progress of each subsection of the project will be compared to the estimates used in the cost summary.
- If a subsection's progress is moving slower than anticipated, action will be taken early to increase the development speed.
- If the action is ineffective and the subsection takes longer than anticipated, the cost of the extra development time will be taken from the buffer.
- If the cost buffer is exceeded, the client will be contacted and an agreement will be made between the project team and client to either drop a lower-priority feature, or increase the development time and thus cost to make up for it.

8 Testing Plan

The project has placed a high priority on constant and consistent testing, especially with regard to our primary tests - usability and reliability testing.

The project group has placed the highest priority on ensuring that, in particular, software reliability and stability is of the best possible quality, and that the software is of a high usability. We placed these at a higher priority due to the nature of the application as a work tool, which we see as requiring ease of use and reliability over other considerations such as performance and aesthetics - though these will also be a focus of the project group.

Another large consideration for the group is security. Although the primary database will be hosted online on a server provided by the client, data will still be stored in a local setting on the clients' machines.

8.1 Test Plan Objectives

The primary objectives of the testing plan are, from a broad perspective, to increase the quality of the application. The areas with which the testing plan will directly intend to increase the quality are:

- Usability and user interface quality will be a primary focus in testing. We want our software to be easy and nice to use. Efficiency will also be a primary target for testing. The application should most definitely be quicker to use than the previous spreadsheet method the client was using. Improving aesthetics will also be a (secondary) goal here.
- Application performance. Part of being nice to use is also responding consistently and responding quickly. We want a minimum of wait times.
- Improve reliability and stability. The program shouldn't crash, eat user data, glitch out, not respond to user input, or do anything to otherwise get in the way of what the user wants to do. Ideally the program should be able to run day in day out without any issues arising.

- Documentation quality. All associated technical manuals, user manuals, packaging, etc. should aim to be free from spelling mistakes, grammatical errors, and other problems. Testing will also tell us if we need more pictures or graphics to simplify instructions, or if they are poorly worded or explained for the target audience.
- Increase the security of the application. Testing will aim to check encryption methods, input sanitisation, and other means of getting to or modifying user data.
- Flexibility. Our client will need to be able to make flexible queries into the database. Our application should be as flexible as the client requires, without sacrificing UI clarity.

8.2 Testing Strategies

Black box testing will be used to test the usability and user interface of the program, and in a more limited sense, the reliability, stability, and performance of the system. To a small degree security testing will also be an objective of black box testing, mainly with regard to user registration and login - we don't want users gaining access to data they shouldn't have access to. Black box testing mainly involves people other than the software developers themselves testing the user-facing side of the application.

White box testing will be used for a more in-depth analysis of the code itself. This will include integration and unit testing. White box testing will mainly involve the project's software developers doing internal tests on the functions and classes of the program.

In either case the good and bad points of the system will be noted, and extra focus made to increase the poorer elements of the product - consistency is of a high concern.

8.3 Test Cases

The following methods of testing the application have been put in place by the project team to ensure a high quality product is delivered to the client, with no unexpected issues or problems.

- Usability testing will primarily be an end-user test case, and will focus on visual efficiency, aesthetics, general usability, and clarity. This test case aims to make the application more approachable and, in general sense, better to actually use.
- Performance testing will be run on multiple systems, especially systems which will be a common use case for the software - older laptops and less capable office computers. The loading times, and general responsiveness of the application will be the major focus of testing performance.
- System testing will be performed and will focus on whether the application works correctly on associated operating systems - all modern Windows operating systems - and whether the software will work correctly on the common hardware in use by the target audience. This includes any potential compatibility issues with other third-party software, and any issues that may arise from differences in hardware.
- Security testing will be done over the course of development to ensure the protection of potentially sensitive client data. This includes any privacy issues associated with that data, but also the protection of the data itself. Tests will be done to try and "break" the program, and cause it to destroy or divulge data that it shouldn't.
- Documentation testing is used to ensure that all documentation relevant to the software project is current, complete, and easy to follow. This will primarily be a job done the the project's editors, with input from the developers. Members of the target audience will also be shown the documentation, to test whether it is easily understood and followed.
- Unit testing will be performed on a functional and class basis, and will be performed by the programmers during development. The primary objective

of unit testing is to ensure the proper operation of software code, and to try and ascertain if any unexpected input or operation could cause issues. Any bugs found during unit testing will be added to the Redmine issue tracker to be fixed by the software developers.

- Integration testing is the final testing case employed by Chimera software and will focus on whether the associated technologies used by the application are working together as expected. This includes any issues that may arise in the database systems used by the application, and any other third-party libraries and middleware used by the developers.

8.4 Test Schedule

The following table lists when the testing will be performed, who will perform it, and a rough idea of the sort of tests each type of testing will entail.

ID	Type	Description	Tester	When (Weeks)
1	Usability	Is the program clear and easy to use? Are the fonts and colours acceptable? Is it aesthetically pleasing without being distracting?	Independent	Weeks 8 / 10 / 12 / 14
2	Performance	Is the program responsive? Does it complete all operations quickly even on modest machines?	Internal and Independent	Weeks 8 / 10 / 12 / 14
3	System	Is the program stable? Does it work on all current Microsoft operating systems? Is there any notable compatibility issues with third-party software (antiviruses etc.)? Any problems with common hardware?	Internal and Independent	Weeks 8 / 10 / 12 / 14
4	Security	Test the security of the application - is client data secure? Are we safe from SQL injection? Are we using safe and secure connection protocols?	Internal (Programmers)	Weeks 8 / 10 / 12 / 14
5	Documentation	Is the documentation easy to follow by the target audience? Are there spelling or grammatical problems? Is there enough diagrams and pictorial instruction to make things clear and easy to understand?	Internal and Independent	Weeks 8 / 10 / 12 / 14
6	Unit	Do all functions work with unexpected (and expected) inputs? What happens if invalid data is entered? What happens if no data is entered?	Internal	Weeks 8 / 10 / 12 / 14
7	Integration	Do all the database connects work properly? Can the tables be downloaded / uploaded / synced? Do all third-party libraries work correctly?	Internal	Weeks 8 / 10 / 12 / 14

Table 13: Product documentation and descriptions

It should be noted that the weeks selected for the above schedule were based on when the builds for our project are being done - on a fortnightly basis starting from the signing of the initial client contract. The 4th build, performed in week 14, will be the build given the most care, consideration and time. This is because the 4th build will be the final build intended to be handed to the client, and thus must pass all associated tests.

8.5 Appendix

Change Control Form

Change Request #: _____		Project: _____	
CHANGE REQUEST INITIATION: Originator: _____ Phone#: (____) _____ email: _____ Date Submitted: ____/____/____ System/Product/Service Name: _____ Version Number: _____			
CONFIGURATION ITEM:		Software: ____ Firmware: ____ Hardware: ____ Documentation: ____ Other: _____	
CHANGE TYPE: New Requirement: ____ Requirement Change: ____ Design Change: ____ Other: _____			
REASON: Legal: ____ Market: ____ Performance: ____ Customer Request: ____ Defect: ____ Other: _____			
PRIORITY: Emergency: ____ Urgent: ____ Routine: ____ Date Required: ____/____/____			
CHANGE DESCRIPTION: (Detail functional and/or technical information. Use attachment if necessary.) Attachments: Yes / No			
TECHNICAL EVALUATION: (Use attachment to explain changes, impact on other entities, impact on performance etc.) Received By: _____ Date Received: ____/____/____ Assigned To: _____ Date Assigned: ____/____/____ Type of Software/Hardware/etc. Affected: _____ Modules/Screens/Tables/Files Affected: _____			
Documentation Affected:	Section #	Page #	Date Completed
Requirements Specification	_____	_____	____/____/____
System Design Specification	_____	_____	____/____/____
System Test Plan	_____	_____	____/____/____
Training Plan	_____	_____	____/____/____
User System Reference Manual	_____	_____	____/____/____
System Maintenance Manual	_____	_____	____/____/____
Other (Specify)	_____	_____	____/____/____
TIME ESTIMATES to make the change: (Use attachment if necessary.)			
Lifecycle Stage	Est. Time	Act. Time	Date Comp.
Analysis/Design	_____	_____	____/____/____
Coding/Testing	_____	_____	____/____/____
Acceptance	_____	_____	____/____/____
Total Hours:	_____	_____	_____
Project Impact Analysis Needed: Yes / No (If yes, include impact on budget, resources, schedule, risk etc.)			
APPROVALS: Change Approved: ____ Change Not Approved: ____ Hold (Future Enhancement): ____			
1. Signature _____		Date: ____/____/____	
2. Signature _____		Date: ____/____/____	
3. Signature _____		Date: ____/____/____	

References

- [Schwalbe, 2011] K. Schwalbe, *Information Technology Project Management 6th Ed.*, Thomson Course Technology, 2011.
- [Emprend Inc, 2006] Emprend Inc, *ProjectConnections.com Template*, Emprend Inc, 2006.
- [PHE.com] PHE, <http://www.phe.gov/about/amcg/toolkit/Documents/risk-management.pdf>, PHE, PHE.