|  |
| --- |
|  |
|  |
| Date |
| Caroline’s Robots |
| Class Room Manager |
| Master Engineers |
| Authored by: Brad Kent |

****

# Caroline’s Robots

## Class Room Manager

Brad Kent

**Executive Summary -** The client Caroline, had requested that a data base + robot to aid and manage day to day in a primary school environment.

Request: To build a custom database to store all relevant classroom activity’s, it will communicate with the school system, this is due to the government regulations already in place. The client also requested that a robot be included to aid in the management of the class. This robot will have communication to both students via activity’s and will report all relevant info to the teacher, then to the database for storing.

This document identifies the input and output interface requirements of the software and how the program will be partitioned on the school computer/ network system.

Caroline will be able to access the database and robot controls via her laptop and/ or tablet. They will allow her to record, submit and retrieve her data to aid in the teaching of subjects to a class with the help of 1 or more robots.

This technical document has been created to aid in any user/’s/ admins, etc. in managing, navigating and understanding the systems process to maintain an effective and healthy workflow.

This system is estimated to require 8 months from interview – > final product, and would cost $40, 000 + Robot Pricing. (see email from manager with detailed pricing, i.e. developers: 10, $30hr, etc.)

**--------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Versions:**

Many drafts and prototypes have been created in the development process they will not be included.

Versions start at 1.0, any other updates will increment by 0.1, i.e. 1.1, 1.2 etc.

You will be notified of any updates and any update will include a technical document detailing the updates made and any changes to how the system operates.

**--------------------------------------------------------------------------------------------------------------------------------------------------------------------**

Signoff

This Technical Requirements Specification has been ratified and accepted by:

|  |  |  |
| --- | --- | --- |
| **Development Organization Representative**  Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | **Client Representative**  Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# 1 Introduction

This specification has been created as part of the contract requirements. Caroline has requested a management system plus a robot would aid her in her daily classroom activity’s. She made requests to the principle of said school, who then informed the state government education body. All legal/ legislative issues where then promptly solved. This request was minute on June 30, 2002.

**--------------------------------------------------------------------------------------------------------------------------------------------------------------------**

*This document covers the Technical Requirements Specification for the utility requested, notably the operational, data and input and output requirements.*

**--------------------------------------------------------------------------------------------------------------------------------------------------------------------**

The objectives of this document are to:

* identify the development and system management standards
* clearly document the data flow for the system,
* present a data dictionary for the system
* identify performance, security, performance, backup, and data maintenance issues
* identify the various roles and responsibilities for managing the system
* present the data storage requirements of the system
* clearly document the user interface requirements.
* provide sufficient information for accurate quotation and development by contact programmers
* provide sufficient information so the technical writers can begin creating the user manual and associated documentation.

# 2 Project Management Summary

## 2.1 Timelines and costing

Arrangements will be made with a contract programmer to complete the development of Caroline’s Classroom Robot/’s.

It is expected that the programming will be finalized and delivered to the Local Library for internal testing within 8 months from the conclusion of formal contract execution. The estimated cost of the project is cost $40, 000 + Robot Pricing

The user and associated technical documentation will be produced by the development group and will begin at the same time as the programming. It is anticipated that this will be completed within 1 week after the software has been tested internally by the Local Library.

## 2.2 Development Methodology

There are many development software development methodology’s, the main 4 are the waterfall model Rapid Application Development (RPD), Joint Application Development (JAD) and the spiral model.

The methodology that I would recommend is JAD, because it involves the client or [end user](http://whatis.techtarget.com/definition/end-user) in the design and development of an [application](http://searchsoftwarequality.techtarget.com/definition/application), through a succession of collaborative workshops called JAD sessions

Spiral, water fall methods etc. mainly involve a very experience team that can make their own decisions based on the project, i.e. if they decide that the system would be beneficial to add other aspects etc. they would need to contact and inform the client and seek approval.

## 2.3 Programming Standards

For the project to proceed succeed and to meet the client’s requirements an initial agreement on how, why, how often etc. the client and company’s repressive should meet/ communicate to discuss/ inform each other of the progress, issues etc. of the application.

Firstly, the dev team and manager should be on the same wave length so to speech, next is to proceed with the technical interview with the client, at beginning ask if client has ever worked with a tech team before and if so doe he/she have any standards in regards to the application. The manager of the dev team should already have instantiated a development methodology and the team should be verses in said methodology.

Programing standards can range from how, when, lengths of the code, to how and when the team tackles problems, to budget and time. All to give the project optimal chances of a successful launch.

# 3 System Considerations

## 3.1 Security Issues and Responsibilities

Both the student and library staff versions of the interface are to be accessible via a database screens (see screen designs) on Microsoft Windows and or Mac. The database & robot program will load Microsoft Access via a security password dialog, and present a program menu within Microsoft Access.

The application will have 3 main layers of access, highest (root) would be the Admins (Tech) for maintaining and upgrading the code, Admin (Office) will have control over all the data in the database, and Caroline will have access to her database profile and will be able to control the robot and its workflow.

With varied password access, Schooling staff can access the full program functionality from any PC holding the interface portion of the database. Students will not be allowed access to underlying tables, relationships or form design, properties, or code. Students will not be allowed to add, edit, or delete any database entries.

## 3.2 Performance Issues

The application is designed to grow as increasingly data and users are added. To achieve the most efficient and most scalable product available the system will include multiple sorting, storing, searching etc. algorithms. The user count would be based on the current schooling size and would be scalable base on said areas statistics of growth.

The main bottle neck will be at school starting time and finishing time as well as any special events. This is due to many actions happening at one point in time and all need to be handled correctly

## 3.3 Networking Considerations

The application will run on the current schooling system, as this is already in place and is already funded and maintained by staff on site. The network will most likely be slower during bottleneck time zones, this must be taken into consideration. The application will be running on a computer/ tablet in a class room and will be communicating with the server over the stated network above.

## 3.4 Hardware Requirements

The hardware requirements would include all server related i.e. networking cable, server hardware, switched etc. all applications will be running on the individual applications with individual users. A simple recommendation would be for ant hardware and software to be no older than 6 years. This is to avoid crashes, glitches etc.

The minimum specification for all devices running the application would need at least 1.2ghz Cpu, dual core, 4 gig ram, HD screen.

## 3.5 Operating System Requirements

The ability to install, run, operate, and communicate wired and or wirelessly. They also must have the technical requirements both hardware and software to do the mentioned and be able to communicate i.e. transfer the data from the application to the server.

They will need to last at least 2years, should be able to operate daily, should also be also can not hinder the workflow of the users. Tablets and desktops/ laptops use different OS’s, this could prove troublesome due to compatibility.

## 3.6 Other Supporting Software Requirements

Other software requirements should include what is considered safe for the client’s systems. The main issues would revolve around either the programs or their output not being compatible with the application. E.g. a document used to record and maintain students records output should be able to communicate with the application.

Tablet and smart phone are the main issue due to them not being able to install and run a desktop operating system, due to this limitation, most software would not be able to run/ operate.

A solution could be to output said programs data to a converter and then to a compatible text file.

## 3.7 Development Environment

A list of all the software, system, hardware, server’s, OS’s etc. this is needed to ensure the dev team can design and optimize the application for the target systems. The dev team should have all the necessary equipment to be able to development and optimize the application. The dev team’s skill sets should reflect the applications technical specifications as well as the testing of the application.

## 3.8 Backup and Recovery Strategies and Responsibilities

All data should be backed up period. The ways and depth of the backups would depend on the client’s system capabilities and needs. A full back up every day, every hour and keeping all backups since the system has launched would be considered overkill and wasteful of resources. Instead a comprehensive overview of the system and what/ how the data is being stored should be the basis of the backups.

I would recommend a full backup of the vanilla system for a complete reinstall in case of total failure. Then a full backup of all data modified, added etc. done at the end of the day after the students leave. Only 6 iterations of said backups should exist. Also, the data should be backed up at an offsite server house (Cloud) run by the school’s management.

## 3.9 Data Integrity and Validation

Due to databases GUI's and how us as people input data i.e. computers cannot predict what will are going to input. E.g. the user got their info through a process that then needed to be recorded i.e. Talking to students etc.

Because of this limitation on a limited number of fields can be checked for their integrity. E.g. a address field can be looked up to see if it is relevant i.e. if the school in in Caboolture, Qld, it would not be possible for an any staff, teachers, students etc. at the school to have address, phones in a different state. The database can though will the aid of either plugins, scripts etc. can prevent duplication, wrongful quires, modifying of data (over writing) etc.

Due to the nature of the limitation and many users some fields with data looking up, syntax (int cannot be string etc.) not all fields will be correct. Though the process of checking and syntax will limit the errors greatly some user checking is still required.

## 3.10 Managing The System - Roles And Responsibilities

3 Main/ Daily Roles:

- Tech Admins – Look after all tech for database & school network

- Office Admins – All data and fields for the database and its users

- Teachers – using and recording related data i.e. classroom - students

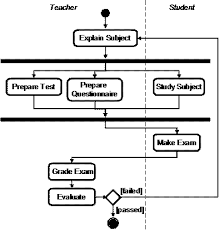
- Possibly Students – students could possible link their profiles to the database

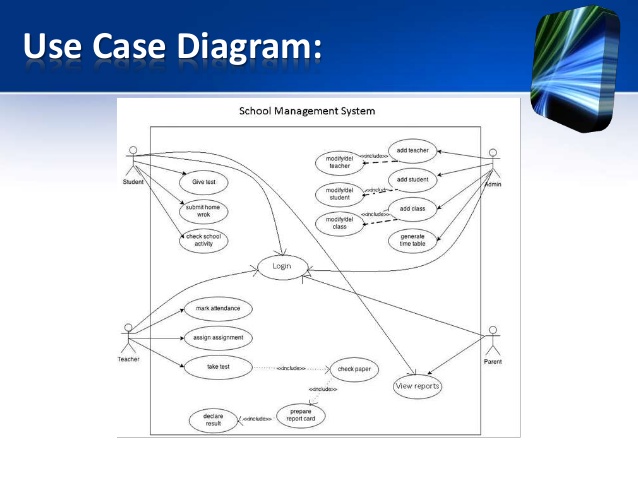
General GUI for teacher – just able to modify permitted fields (Students, room etc.)

Detailed and compact GUI for office admins – able to select and modify whole class, teacher groups etc.

Full access for the tech admins, access to all gui’s for problem solution searching. They will also have access to backups, refreshes etc.

# 4 Data Flow and Data Dictionary

-----------------------------------------------------------------------------------------------------------------------------------------



# 5 Data Requirements

## Normalization:

Un-Normalized Data:

{Class ID, Class, Room, Block ID, Start Time, Finish Time, Note, Teacher ID, Teacher First Name, Teacher Last Name, Teacher Email, Teacher Phone {Student NO, Student Name, Attendance, Note}}

1st Normalization – Remove Repeating Groups

{Class ID, Class, Room, Block ID, Start Time, Finish Time, Note, Teacher ID, Teacher First Name, Teacher Last Name, Teacher Email, Teacher Phone}

{Class ID, Student NO, Student Name, Attendance, Note}

2nd Normalization – Remove Partial Dependences

{Class ID, Class, Room, Block ID, Start Time, Finish Time, Note, Teacher ID, Teacher First Name, Teacher Last Name, Teacher Email, Teacher Phone}

{Class ID, Student NO, Attendance, Note}

{Student NO, Student Name}

3rd Normalization – Remove Transitive Dependencies

**Teacher**: {**Teacher ID,** Teacher First Name, Teacher Last Name, Teacher Email, Teacher Phone}

**Class**: {**Class ID**, Class, Room, Block ID, Start Time, Finish Time, Note}

**Attendance**: {**Class ID**, **Student NO**, Attendance, Note}

**Student**: {**Student NO**, Student Name}

***Teacher – Class \_ Attendance – Student***

Un-Normalized Data:

{Course ID, Course Name, Course Syllabus, Course Textbooks, Course Grading, Course Assignments {Assignment ID, Assignment Name Assignment Due Date, Assignment Time -Frame, Assignment Template, Assignment Task Sheet} {Activity ID, Activity Name, Activity Category, Curriculum, Activity ETA Time - Frame {Task ID, Activity ID, Robot ID, Page NO, Part NO, ETA Time, Marking}}}

1st Normalization – Remove Repeating Groups

{Course ID, Course Name, Course Syllabus, Course Textbooks, Course Grading, Course Assignments ID, Sub-Category ID, Course ID, Activity List ID’s, Activity List ID’s, ETA}

{Assignment ID, Assignment Name Assignment Due Date, Assignment Time -Frame, Assignment Template, Assignment Task Sheet}

{Activity ID, Activity Name, Activity Category, Curriculum, Activity ETA Time – Frame, Task ID, Activity ID, Robot ID, Page NO, Part NO, ETA Time, Marking}

2nd Normalization – Remove Partial Dependences

{Course ID, Course Name, Course Syllabus, Course Textbooks, Course Grading, Course Assignments ID}

{Sub-Category ID, Course ID, Activity List ID’s, Activity List ID’s, ETA}

{Assignment ID, Assignment Name Assignment Due Date, Assignment Time -Frame, Assignment Template, Assignment Task Sheet}

{Activity ID, Activity Name, Activity Category, Curriculum, Activity ETA Time - Frame}

{Task ID, Activity ID, Robot ID, Page NO, Part NO, ETA Time, Marking}

3rd Normalization – Remove Transitive Dependencies

{Course ID, Course Name, Course Syllabus, Course Textbooks, Course Grading, Course Assignments ID}

{Sub-Category ID, Course ID, Activity List ID’s, Activity List ID’s, ETA}

{Assignment ID, Assignment Name Assignment Due Date, Assignment Time -Frame, Assignment Template, Assignment Task Sheet}

{Activity ID, Activity Name, Activity Category, Curriculum, Activity ETA Time - Frame}

{Task ID, Activity ID, Robot ID, Page NO, Part NO, ETA Time, Marking}

Teacher – Class – Course – Sub Categories – Activities – Task

Student – Class – Course – Sub Categories – Student Activities – Student Tasks