|  |
| --- |
| Software Design and Development |
| Major Project: Classroom Manager |
| Analysis and Design Report |

|  |
| --- |
| Euan Mendoza  2-3-2019 |
| Version: 0.3.0 |

Table of Contents

[Introduction 2](#_Toc535762261)

[Client Needs 2](#_Toc535762262)

[Requirements 2](#_Toc535762263)

[Functional Requirements 2](#_Toc535762264)

[Non-functional Requirements 2](#_Toc535762265)

# Introduction

This is a simple classroom manager with a focus on security, and privacy.

The project uses the MIT License.

# Client Needs

The client needs are as follows:

This program must allow a teacher to manage the day-to-day classroom tasks, which must include.

* class lists
* Recording of marks
* recording of timetable
* Behaviour notes

# Design Objectives

* To be easy to use with an intuitive user interface
* For each component to ‘imply’ what it does (use of labels and icons)
* To be safe and secure for educators and students alike
* To be type safe
* To be able to store data in a database
* To be responsive and fast
* To run without bugs

# Requirements

## Functional Requirements

This software will allow Users to:

1. Have multiple users
2. Have a system admin account with the ability to add or remove students
3. Sign in with Microsoft Outlook
4. Have a list of students sorted by last name alphabetically
5. Record student information
   1. Have the required fields of student names and year group
   2. Non required information Contact information, Parent contact information, Student ages, Color house, Medical information
6. Keep a list of classes with a list of students attached to each class
7. A user can be assigned to a certain class
8. A user can mark the role of their assigned class
9. A user can assign tasks, and behaviour notes to their assigned class
10. A user can assign tasks, and behaviour notes, and mark tasks to a student in their class individually
11. The ability for an admin to add or remove, users, students, other admins, terms (default 4) with term begin and end dates (default spx term dates), the ability to add periods per standard school days (default 6) with a lunchtime and recess.
12. Save data so that it is persistent and secure

## Non-functional Requirements

The following is a list of requirements that do not relate directly to the business functions of the software

1. Password and username authentication, for security
2. Passwords will be hashed and stored rather than storing the actual password
3. Data will be encrypted and stored into a sqlite3 database
4. A 256 bit security key will be used to encrypt the data
5. Text information will be saved as encrypted plain text but able to be outputted as rtf files
6. The program will be a deployable webserver, the backend and the frontend will be deployable at the same time but written separately.
7. The program is packaged using webpack.js and build using the yarn build system
8. The program will use a Go backend Graphql API
9. Original code will be used, although many common algorithms will be implemented and with the exceptions of the following libraries
   1. Materialize UI
   2. Typescript
   3. Webpack.js
   4. React.js
   5. Random number generation
   6. Go-graphql and Apollo client
10. Standard Algorithms used are
    1. Text parsing
    2. XOR Cypher Algorithm
    3. Merge Sort
11. Follows material design standard

## Compatibility

Needs to be able to work on Microsoft Edge as the baseline.

Google Chrome for phones.

## Performance

Needs to be able to have a suitable response time, less than 2 seconds after a button is pressed

Needs to run on 64-bit servers

## Boundaries

Windows operating system for server side deployment.

A variety of

Small screen, large resolution high dpi screens.

## Project Plan

# Design Specifications