Planning

Team Leader: Alex L
Team Collaborators : Alex C, Jone S, Ritika R

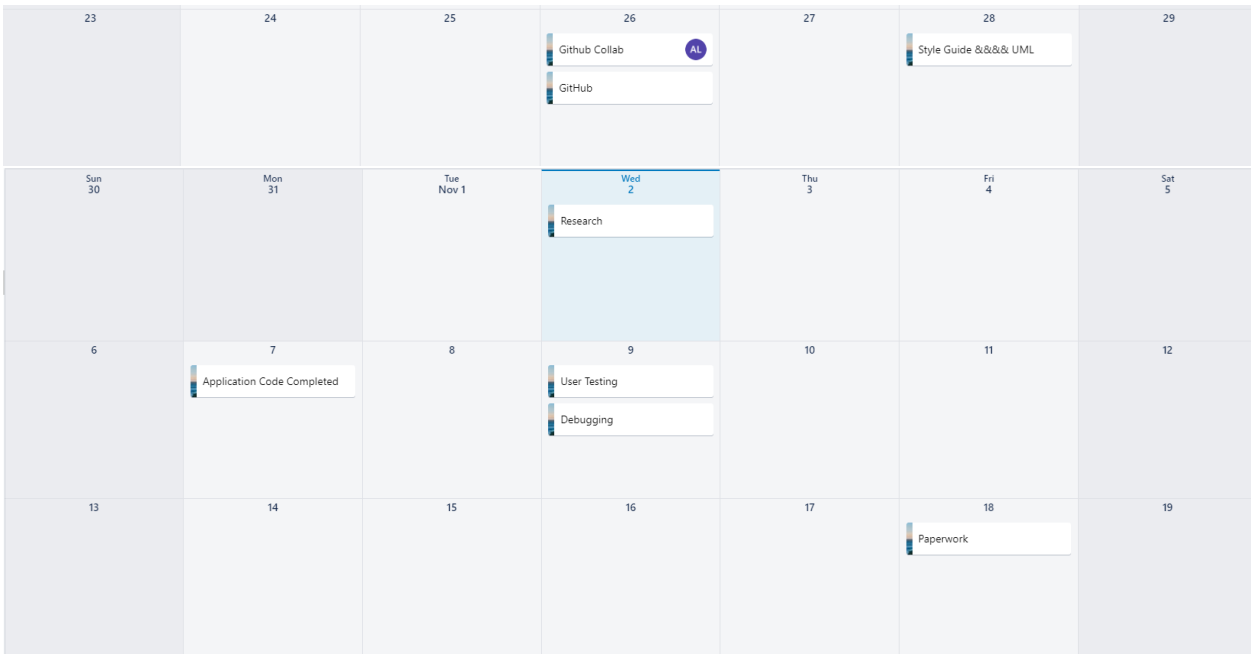
Team Leader Responsibilities:

- Ensure the group stays on the timeline and tasks are completed on time.
- Ensure the group has the information they need to complete tasks and knows what to do.
- Ensure all issues are dealt with and resolved.
- Make any decisions the team can't conclude together on.
- Keep the team informed, and allocate tasks as needed.

Team Collaborator Responsibilities:

- Keep the team informed about what they are doing and of their absence.
- Adhere to the timeline and get tasks done in a reasonable time.
- Post updates in the blog (Teams) and updates workflow (Trello) as needed.
- Seek help or guidance when unsure what to do.

Timeline:
As a team, we came up with a timeline for everyone to adhere to -



We have 4 weeks for the project and with the timeline set, we started delegating tasks as required to team members.

** See Trello Evidence in Collaboration Evidence to see Specifically allocated tasks. However, tasks were evenly allocated for all team members to work on.

Research

Technical Options

Language

There are many coding languages out there, many have their pros and cons:

Language	Description	Pros	Cons
C++	C++ is a general-purpose language used in systems, GUIs, and embedded software.	Good memory management. Very fast	High learning curve. Manual memory management
C#	C# is an object-oriented, modern language	Fast. Automatic memory allocation and garbage collector	Average performance. Relies on the .NET framework
Java	Java is an object-oriented, and multi-platform language	Simple, Automatic memory allocation and garbage collector	Slow, uses more memory
Python	Python language is used to make websites, and software, automate tasks, and data analysis	Flexible and Extensible. Beginner friendly	Security. Bad garbage collector. Code design

We picked C++ because that is a language our entire team is comfortable with. It is one we are learning and know we can all code in whereas we haven't all tried other languages. Picking one we all know is a good option as we need to be able to work together.

IDE

IDE	Description	Pros	Cons
Visual Studio Code	VSC is one of the most popular free open-source code editors by Microsoft. Provides many extensions including C++.	Includes Git Support Syntax highlighting Auto-code completion extensions	Glitches with certain plugin combinations. High learning curve.
Code::Blocks	Open-source and free IDE for C and C++ used on multiple platforms, and built-in C++ using a GUI widget tool called wxWidget	Cross-platform Fast performance Extensible with plugins debugger	manual setup for the GNU compiler. Looks outdated. 0 code suggestions
CLion	CLion is a modern IDE by JetBrains which is designed to run C and C++ on various platforms like Windows, Linux, and macOS.	Smart code Completion Provides Refactoring. Built-in tools. Smart Code analysis.	No inbuilt compiler Bad mix with GitHub. runs 64bit systems
Visual Studio 2019	Visual Studio is the most popular choice for good reason: an impressive collection of available Visual Studio extensions. likely the most popular C++ IDE for Windows.	cross-platform. many programming languages. excellent syntax highlighting.	High learning curve Sometimes unresponsive/crash. poor logs for errors. Plugins may not work.

We picked Visual Studio 2019 because for most of the course our group has learned and studied solely using Visual studio code, due to this and the ease with which the IDE works with console applications makes it a simple and perfect choice for this assignment.

Version Control

Version Control	Description	Pros	Cons
CVS	Concurrent Versioning System is a free client-server revision control system in the field of software	Good cross-platform support and fully featured command line tools. Well known.	No merge tracking. Integrity checking commits and poor support for distribution.
SVN	SVN is a free and open-source software as an alternative to CVS that is aimed to be more reliable and bug-free.	Good GUI supports empty directories, better Windows support, easy to use.	Bugs relating to file naming and directory naming. Slower, Fewer commands.
GitHub or git	GitHub is a code hosting platform for version control and collaboration. Let you and others work together on projects.	Easy collaboration between groups. An online repository for backup. Guides and online help.	Hard to revert bad merges. GitHub website downtime affects work times.
Microsoft Teams Files	Microsoft Teams is cloud-based team collaboration software that includes messaging, calling, video meetings, and file sharing.	Variety of options for communication between group members. Easy access to group documents.	Unnecessary storage consumption. Lack of notifications. Confusing merge system/too many tools.

We picked GitHub because our team is most familiar with the operations and merging system implemented by this website/application. The ability to look back through an individual's work and adjust each other's code with branches and eventually merging, singles GitHub out for its simplicity and group collaboration. GitHub fits well for the team as it also links well with our choice of IDE VS 2019.

Collaboration

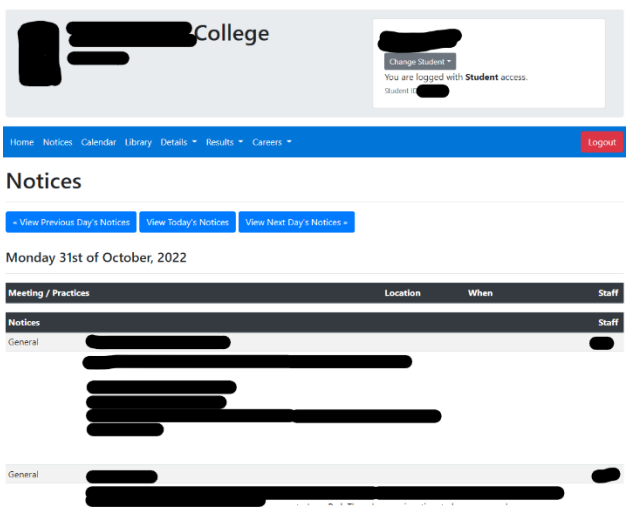
Research collaboration software and explain why you picked it (we are using Trello)

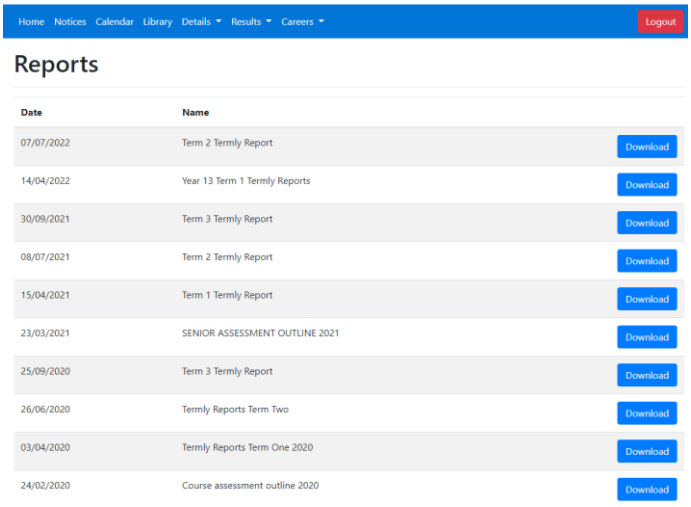
Collaboration	Description	Pros	Cons
Trello	the visual tool that empowers your team to manage any type of project, workflow, or task tracking. Add files, checklists, or even automation	Clear and concise formatting, multiple options for customization. Easily form project guides	Limited storage capacity, the improvement required on commenting feature and editing
Click Up	Click up is a large-scale task manager for large companies to allow workers to see tasks and organize them as needed	Lots of features and good for large-scale companies	Too complex for individual needs. Not free for teams.
monday.com	Workflow software allows you to manage tasks within a team and manage a timeline.	Good project management tools and software	Subscription-paid software for features you would want.

We picked Trello because it is free to use and collaborate with our small team and was simple to use/understand. It will make organizing the workflow of the project easy because everyone in our group already knows how to use Trello.

SWOT analysis of Existing Apps

Kamar -

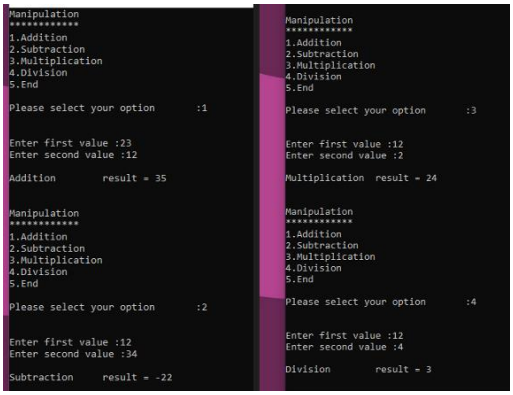




Strengths	Easy to read Menu Bar. The reports list is easy to read and easy to use -> download the report you want. Login/Logout is simple with a timeout feature.
Weakness'	Some links don't work -> library is not set up properly and still showing, notices don't show ALL notices, only some of them.
Opportunities	Small competition for this type of software. Kamar is widely used in NZ. Include tutorials on how to use it, with the feature to never show again.
Threats	No clear instructions on how to use KAMAR, you must learn your way around to find what you are looking for.

Console App – Calculator

Strengths	The information provided is clear and concise, focuses on information and selection of data, little to no clutter of unneeded information. Spacing horizontally is easy to follow with like information next to each other and unlike separated.
Weakness'	Some of the spacing across is not even and hard to follow
Opportunities	Could allow in-line operations i.e., 1 x 2 instead of selecting option 3 and entering 1, 2.
Threats	The console is very restrictive, can't push buttons like a normal calculator.



Console App – Game

```
What game do you want to play?
1 - Tic Tac Toe
2 - Number Guessing
3 - Worldle
4 - Adventure Game
5 - Pong
6 - Hangman
0 - Exit
2

Loading Number Guesser...

1 - Easy    1 - 10
2 - Medium  1 - 25
3 - Hard    1 - 50

0 - Exit

Enter difficulty : 2

OKAY, I have though of a number (:

You have 5 guesses left!
```

```
You have 5 guesses left!

Enter your guess between 1 and 25.
Guess : 25
That is NOT correct...
My number is SMALLER than your guess.

You have 4 guesses left!

Enter your guess between 1 and 25.
Guess : 20
That is NOT correct...
My number is a LITTLE SMALLER than your guess.

You have 3 guesses left!

Enter your guess between 1 and 25.
Guess : 18
That is correct!!!
YOU WIN!!!
```

Strengths	Capitalizing keywords makes readability better. The spacing is easy to follow. The format is consistent and easy to learn/follow.
Weakness’	Too many exclamation marks.
Opportunities	Console – add colours and symbols to increase readability.
Threats	Having to push enter for one-character inputs is a pain but also a limitation with C++ and the console.

Possibilities and Limitations in Console Applications:

- A console application is an app that uses the console as its UI. So, its only limit is that it can only do character-based display.
- The console interface is limited to text input/output. You can't do graphics or sound with that interface. You must use a library to provide a GUI/ SDL to give access to graphics, sound, mouse, and so on such as win32 API.
- A console application cannot display visuals, a C++ console application can process visual files from a library (such as icons, bitmaps, video, and so forth).
- Due to the fact you don't require a GUI you can focus on the language, the data structures, the classes, and the algorithms.

Identify Problems in the Existing Systems:

- Loading times for student information take too long due to the sheer amount of information provided and GUI elements affecting speeds.
- Certain links appear blank or don't work optimally due to failure to properly link the individual tabs and options within the program.
- Pages like demographics and medical information should be easier to manipulate. Certain elements of information within the program require obvious changes to simplify and streamline the user experience.
- Putting together a report that meets your needs is confusing and time-consuming. Unable to directly make changes to reports and other components of information.
- If your internet is not at optimal speed, it's very slow and won't connect. Requires constant internet access to view the already implemented data.
- The look and feel of the software are dated. The data in which it is provided and displayed is boring and hard to read sometimes.
- Being able to pull Reports out takes training and more than a novice approach. General use of the application can be confusing for new users due to the number of different options available and how they are implemented.
- The website can be overwhelming due to the many different tabs and functions available.

User Experience with Existing Apps

Due to the limitations of creating a console program, the user experience will depend upon how we design the simple aspects. From the menus to the submenus, the way we display contents/information in terms of the formatting and choice of wording. The user whether an admin or student/parent should easily understand the prompts that are given to them. This is explained in more detail about the limitation of a console earlier. However, we will also get various teachers and students from a local school to try and use some console applications and determine whether the UI is a good user experience based on real people and our personas. We also performed a SWOT analysis.

Personas:

Lydia

Lydia is a teacher at Upper Hutt College who teaches English101

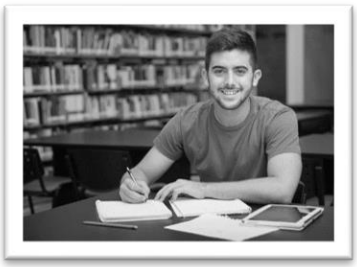
As a teacher, Lydia wants to empower her students to better themselves by appreciating the English language and its beauty. Instead of doing admin duties, she prefers to use that time to better help her students.

Needs:

- Faster attendance completion
- Feedback for students and parents
- Easy to edit report cards
- Clear and concise information formatting
- Quick loading times

Pain points:

- Confusing links
- Unneeded information
- Too many options and links



John

John is a student at Hutt Valley High School who studies Math101

As a student, John requires quick and easy access to his report cards to work and improve on his studies and get feedback from his tutors. Time is his greatest problem as he would rather study than spend time messing around with his report card.

A bonus would be easy access for his parents so they could check up on his work without having to contact John directly, saving extra time and work.

Needs:

- Faster attendance completion
- Feedback for students and parents
- Understandable and readable data
- Simple log-in/sign up

Pain points:

- Dislikes clutter
- Unneeded information
- Lengthy loading times

The outcome of personas:

The main points include a focus on easy access to the application for students, parents, and teachers, short loading times, clear and concise information, and understandable links to the user's desired function.

C++ provides for most of our personas' needs due to the way that the information is shown to the user and its focus on data over UI.

This in turn alleviates the loading times of the application as it doesn't have to load pages with colour, pictures, or other elements that take up processing power.

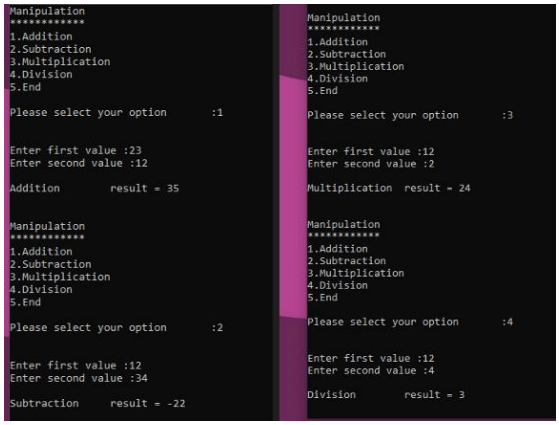
Assumptions (Users):

- All users will understand English
- All users know how to type on a keyboard
- All users would push enter after putting in the text

Assumptions (Console Applications):

- Users would push 'enter' or 'return' when using the application without instruction to do so.
- Users will type help when they are unsure what to do.
- The 'Exit' will typically exit the application
- Help will show a help menu
- The 'Back' will go back to the previous menu
- Options are easier to select when they are numbered for input, i.e., 1 for start.

User Testing of Console Applications:
We asked 5 randomly selected users to represent most users to use console applications and see how they used it to determine if our assumptions were correct or incorrect. We used a calculator console program and asked users to perform the same tasks. We timed the average amount of time each task took and recorded pain points where the user was either unsure or stuck.



Overall Time	Task 1	Task 2	Task 3	Observations
30s	12s	10s	8s	Spacing of output slowed them down when giving us the answer -> readability of console
35s	14s	11s	10s	Wanted to click on the word, quickly typed in the number afterwards
23s	10s	8s	5s	Noticed the longest part was them looking at the menu at the beginning, numbering system was fast for them.
112s	90s	12s	10s	*Didn't press enter for first task for a while -> thought program was loading
33s	13s	10s	10s	No particular / new pain points

Comments by testing users:

- "The words are all the same color, and it makes it hard to read, however the spacing of the options and numbering system makes it better."
- "The numbering system makes selection fast as I don't have to type the word."
- "The console is weird, but I eventually understood how it worked when I had to push the enter key to put in the input."

I noticed that a trend occurred where testers would learn the options available to them after using the program and begin to get faster and faster at using the program.

Outcome of ALL Research:
Most of our assumptions were correct. There were a few things to point out however:

- Users would usually push 'enter' or 'return' when using a console app for input
- Exit will typically exit the application; however, users are more likely to push the X in top right
- Help will show a help menu, although most users won't use it
- Options are easier to select when they are numbered for input, i.e., 1 for start. – Confirmed as it increased speed.

For the console to be readable we need to take our research into consideration with things like a screen layout, spacing, colour if possible, numbered selections for options and the ability to exit at any point.

*****A list of ALL requirements is listed later in this report.*****

Business Requirements

- Add new staff records/edit
- Add new student/edit
- Attendance management
- Faculty management
- Student record management (Student Information)
- Create edit and update school notices

User Requirements

- View academic results
- View school notices

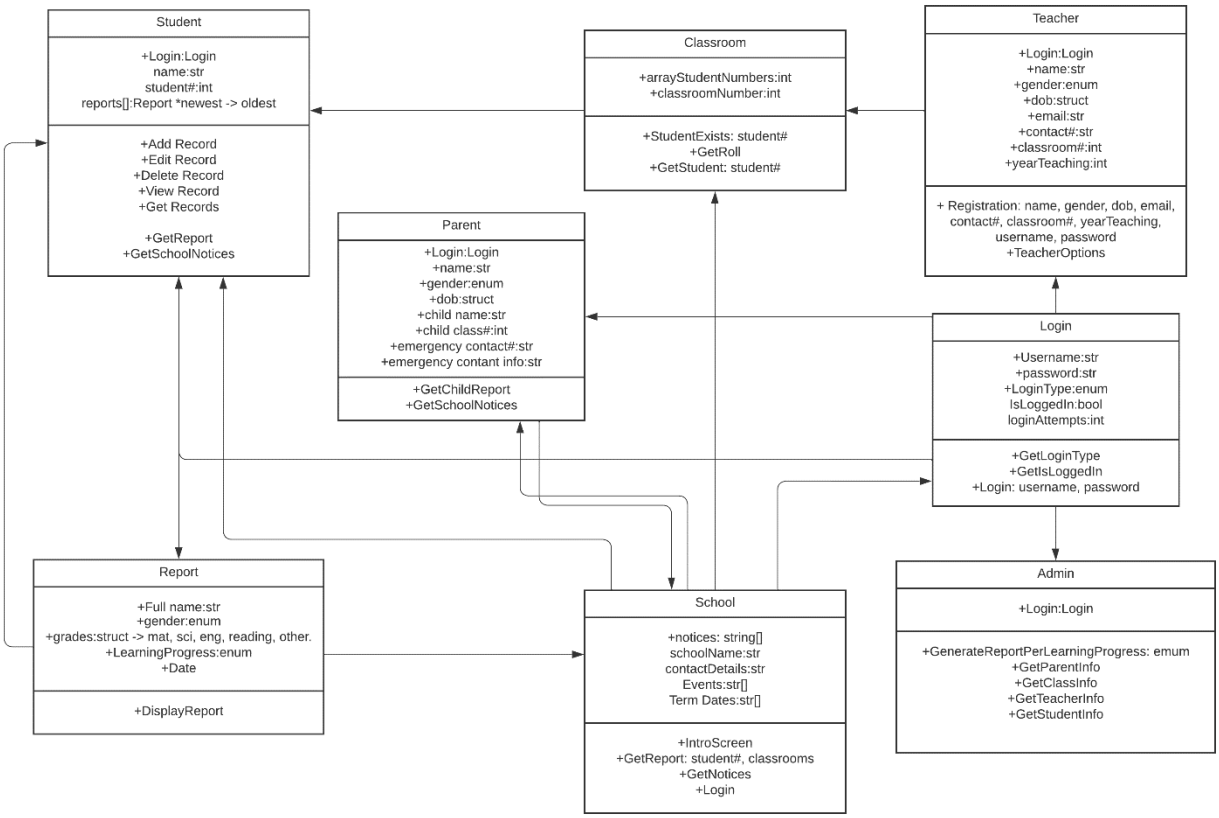
Business wants to be able to add new staff records/information that will help with the management of the faculty, able to display information for specific staff on demand. Upon editing this information, it should be able to update accordingly.

Business also needs to edit and add in new students and their associated records, ranging from attendance, academic results, and subjects that the student is taking. Also want the ability to create and edit students' timetables.

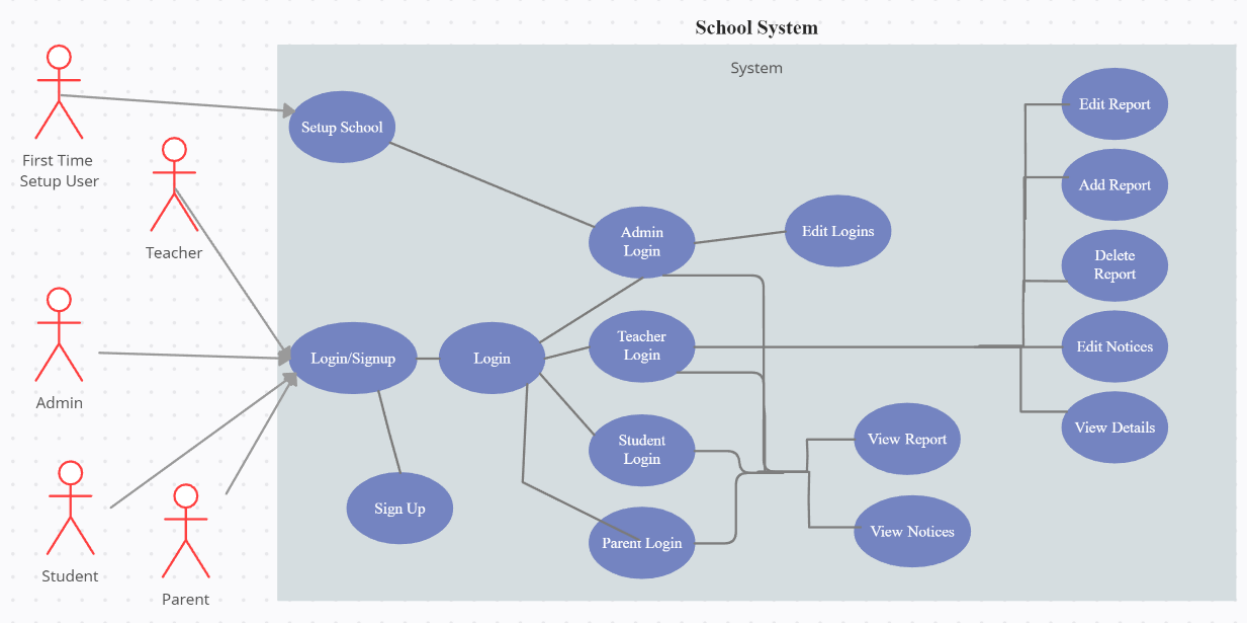
The regular users in this case will be students and parents. For students they will only be able to view certain information. Information that students have access to will be their timetables, results for tests/exams. A parent will be able to view student results, attendance, and school notices.

UML Diagrams

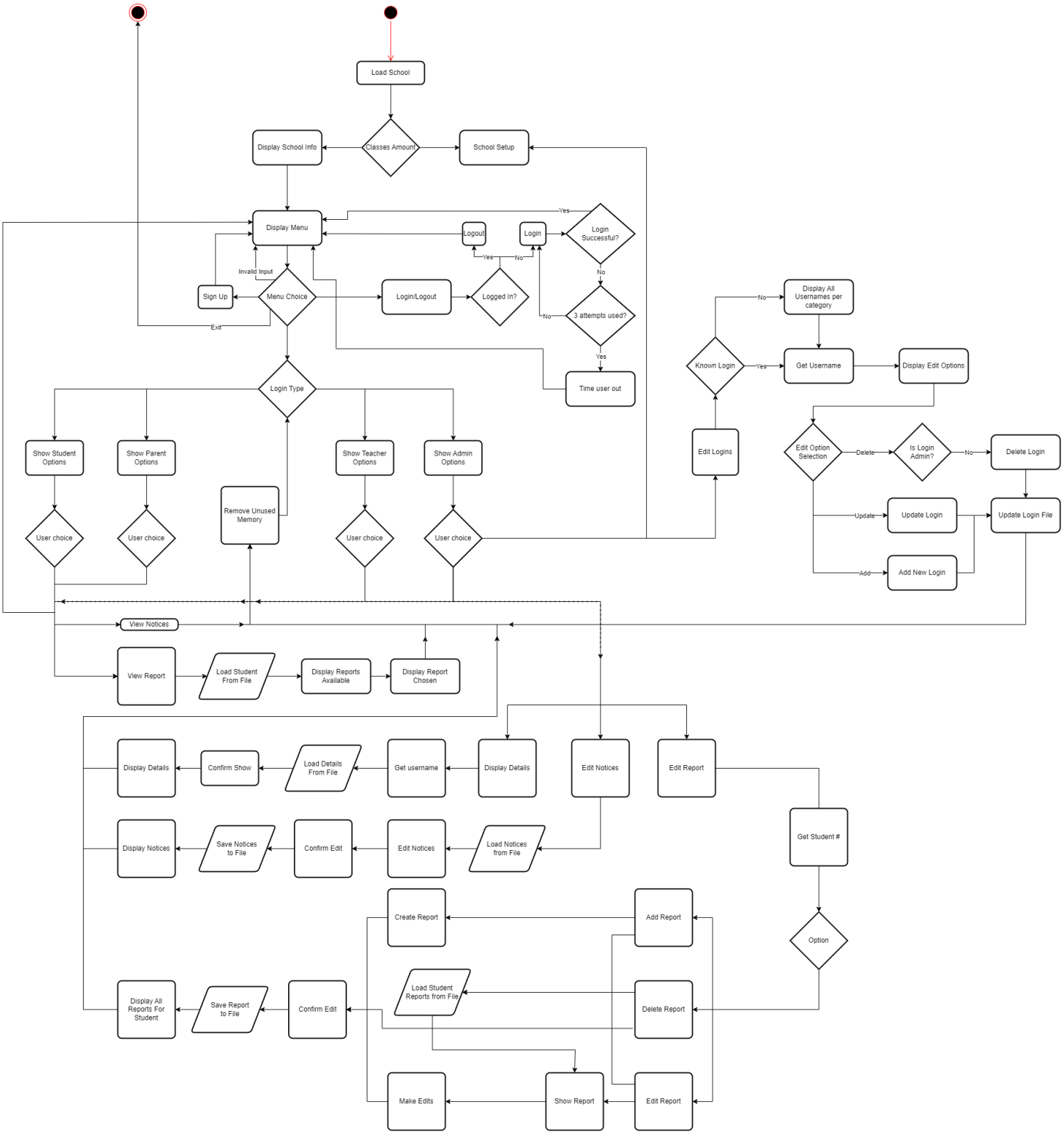
Class diagram:



Use case diagram:



Activity Diagram: (Like a flow chart)



Code Style Guide

C++ Version

This style guide is for C++ 14 Standard. This is the default for any project in Visual Studio so any files added to the project should be checked to ensure the language standard is C++ 14 Standard.

Header Files

Use the C++ header files. A header file with .h is meant for C not C++. For example, you would use cstring instead of string.h as the string header file. Make sure you are using the latest version of a header file. Only include header files you need for your code to function.

Naming Convention

Function names will follow the camelCase naming convention. -> camelCase

```
void getInput() {
```

Local variables will follow the camelCase naming convention. -> camelCase

```
int varaibleOne
```

Block variable will follow camelCase unless it is the same name in which case you would put _ before the variable. -> camelCase then _camelCase
*AVOID _ where possible.

```
string myArray[5];
string target = "test";
for (string element : myArray) {
    string _target; // _ because target was taken
    if (element == target) {
        _target = element;
```

Const variables will be SNAKE_CASE to show their difference. -> SNAKE_CASE

```
const int SCREEN_WIDTH = 300;
```

**** ALL NAMES ARE TO BE APPROPRIATELY NAMED FOR ANYONE TO UNDERSTAND OR COMMENTED**

Static and Global Variables

There will be no global variables within any files to prevent naming conflicts. All variables will need to be accessed either as a parameter or pointer.

Overloading

Overloading will start with the least parameters and underneath follow least to most parameters. Each overload must have a comment to define its purpose even if it's the same.

```
/// <summary>
/// An overloaded fucntion
/// </summary>
void overLoad() {

}

/// <summary>
/// An overloaded fucntion with a paramater
/// </summary>
/// <param name="a"></param>
void overLoad(int a) {

}

/// <summary>
/// An overloaded fucntion with a paramater
/// </summary>
/// <param name="a">null effect</param>
/// <param name="b">null effect</param>
void overLoad(int a, int b) {

}
```

Code layout:
From top to bottom follow -> C++ headers, enum definition, struct definition, class definition, function prototypes, main function, function prototype definitions.

```
// header files
#include <iostream>

// enum definitions
enum enumExample {
    a,
    b,
    c
};

// struct definitions
struct structExample {
    int a;
    int b;
    int c;
};

// class definitions
class classExample {

};

// function prototypes
void FunctionExample();

// main function
int main()
{

}

// function definitions
void FunctionExample() {

}

void proper(){

}

void improper()
{

}
```

Using
using namespace std is bad practice and is not allowed, however to prevent repetition, you can use using std::member where member is what you are using. For example using std::string allows us to write string instead of std::string everwhere. Using statements are to be put under the header files in preprocessor.

Cin
Don't ever use Cin as it has bugs when using reading spaces. Instead use std::getline(cin >> std::ws, varName);

```
std::string input;
std::getline(std::cin >> std::ws, input);
```

Comments
All functions and classes must have /// XML comments

```
/// <summary>
/// I am a XML tag for a function
/// </summary>
void test() {

    // I am a comment for a variable
    int testVar = 2;

}
```

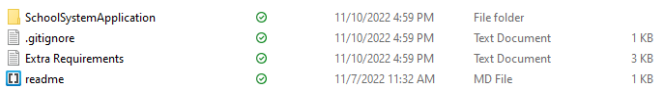
Variables can have // comments above or next to it to define what the variable is used for.

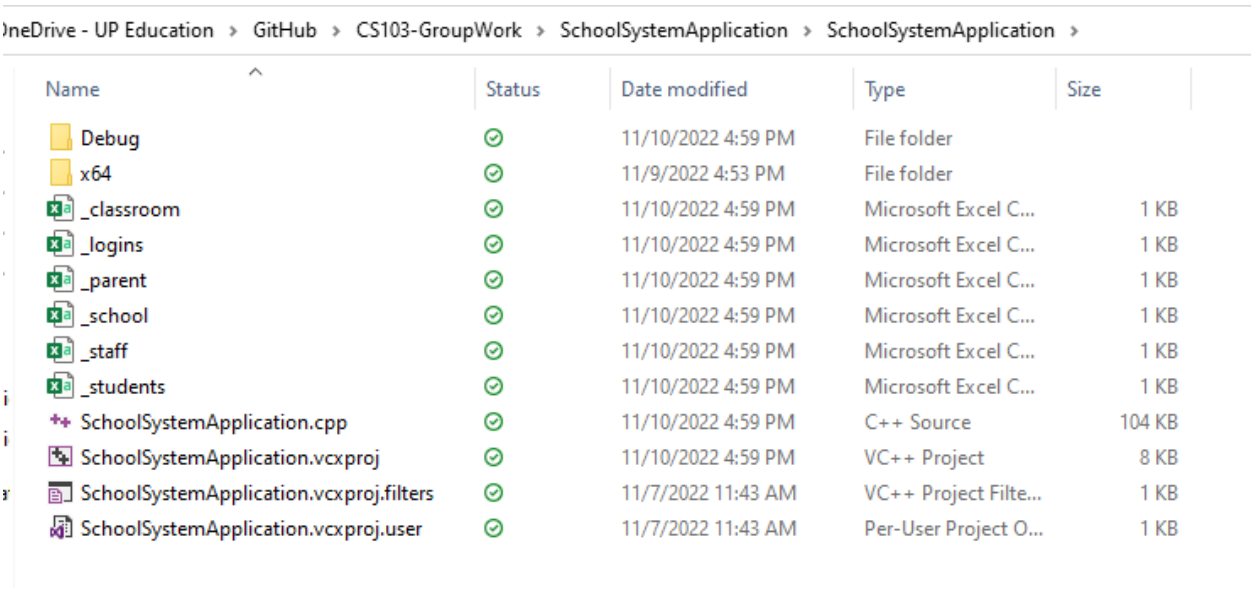
To summarize code use a // comment above it. Add blank lines in between lines of code that aren't similar or don't apply to that comment

Bracket convention
Use C++ style convention with { next to last line and } underneath last line as shown in example

Data Structures
To store data we will use file loading and saving. File saved which contain data will be csv's and stored under the naming convention "_name.csv". Multiple files are allowed for different types of stored data. Data will be loaded to vector<vector<string>> to store data as if it were in the csv format. Data should not be directly edited/read to/from the file instead using the internal memory and then saved/loaded from memory.

File Management
Git will allow for file collaboration, but any coding related files should be in the coding folder and all other files in the main directory:





OneDrive - UP Education › GitHub › CS103-GroupWork › SchoolSystemApplication › SchoolSystemApplication ›

Name	Status	Date modified	Type	Size
Debug	✓	11/10/2022 4:59 PM	File folder	
x64	✓	11/9/2022 4:53 PM	File folder	
_classroom	✓	11/10/2022 4:59 PM	Microsoft Excel C...	1 KB
_logins	✓	11/10/2022 4:59 PM	Microsoft Excel C...	1 KB
_parent	✓	11/10/2022 4:59 PM	Microsoft Excel C...	1 KB
_school	✓	11/10/2022 4:59 PM	Microsoft Excel C...	1 KB
_staff	✓	11/10/2022 4:59 PM	Microsoft Excel C...	1 KB
_students	✓	11/10/2022 4:59 PM	Microsoft Excel C...	1 KB
SchoolSystemApplication.cpp	✓	11/10/2022 4:59 PM	C++ Source	104 KB
SchoolSystemApplication.vcxproj	✓	11/10/2022 4:59 PM	VC++ Project	8 KB
SchoolSystemApplication.vcxproj.filters	✓	11/7/2022 11:43 AM	VC++ Project Filte...	1 KB
SchoolSystemApplication.vcxproj.user	✓	11/7/2022 11:43 AM	Per-User Project O...	1 KB

Screen Layout

**** Screen layouts will go here – note probably after we complete code as it will be easier to show

```
Options:
1. Login
2. Exit
3. Parent Sign Up

Selection : 3

Enter a new parent number : 21
Is your child already enrolled with us?
y or n : n
Enter a new student number : 11
Enter student's full name : StudentTestName

----------------------------------------
---------Create Login : Student---------
----------------------------------------
Enter New Username: studenttest
Enter New Password: studenttest

Login SAVED...

Enter adult's name : ParentTestName
Enter emergency contact phone number : 0123456789
Enter address : 123 Address Street
Enter DOB : 12/12/2012
Enter email : parent.email.test@mail.com

Parent Created...

----------------------------------------
----------Create Login : Parent---------
----------------------------------------
Enter New Username: parenttest
Enter New Password: parenttest

Login SAVED...

Options:
1. Login
2. Exit
3. Parent Sign Up

Selection :
```

```
*****************************************

Welcome to (SETUP SCHOOL NAME)
Contact details:

*****************************************

Select a number from options below to navigate this program

Options:
1. Login
2. Exit
3. Parent Sign Up

Selection :
```

Final Requirements (Specifications):

Introduction screen should include:

- School name
- Contact details
- Login option for all users
- Registration options for parents
- It can also be logged in and controlled by an administration account

New teacher registration screen (only accessible by an admin) should include:

- Should have full name, gender, dob, email, contact number, classroom number, year they teach (e.g., year 1), username and password.

All Logins:

- Should be verified against login usernames and passwords file
- Provide three login attempts to access the system. Otherwise notify them to login after some time.

Admin screen includes:

- Create, Edit and Delete Students, Student's reports, Parents, Teachers, and Classrooms.
- View, Edit, Delete, and Add School Notices.
- Edit School Name and Contact Details.
- Edit ANY Login Details

Teacher screen includes:

- View, Edit, Delete, and Add School Notices.
- View any information on anyone in the system
- Add, Delete or Edit a student's report.

Parent registration screen should include

- Full name, gender, dob, email, contact number, child full name or names, child classroom number, child parent/caregiver full name and the emergency contact number, username, and password.

Student Reports

- Full name, gender, math, science, writing, reading and Others (might be sports, art, music or friendly etc.) and Learning Progress.
- Learning progress could be measured by (Achieved, Progressing, Need Help).

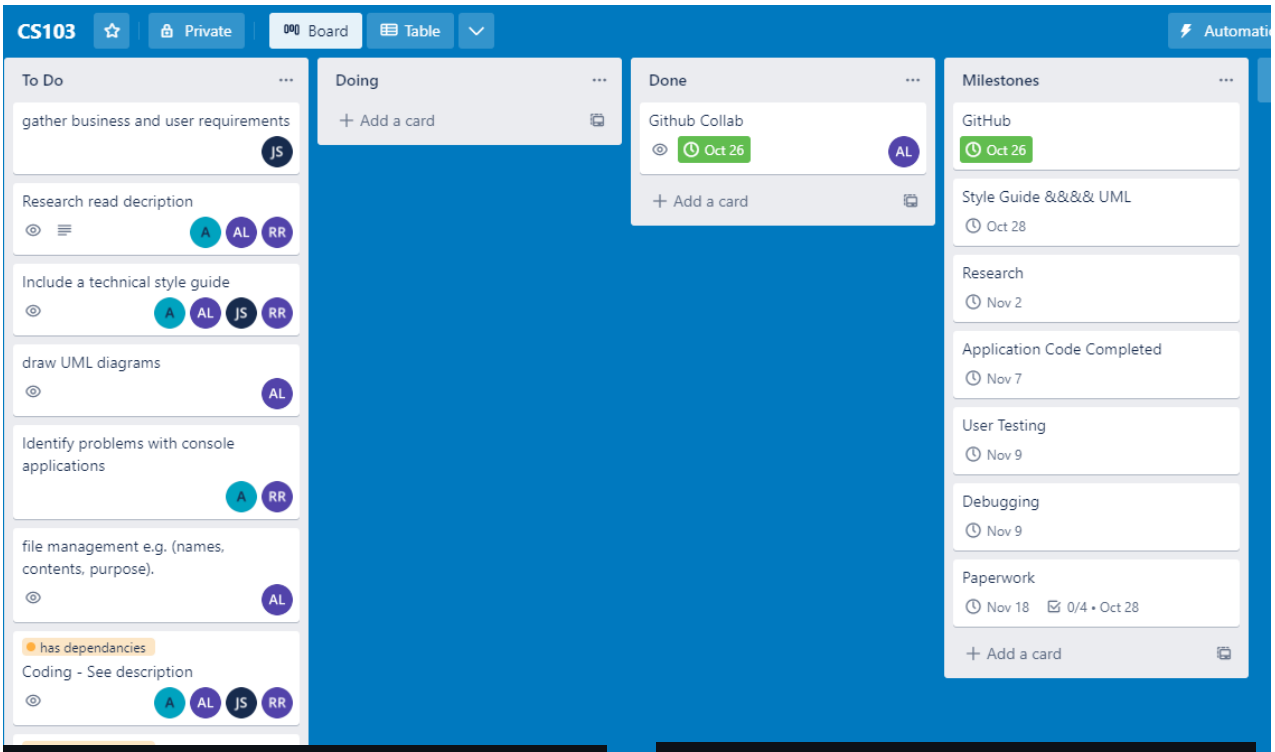
Student/Parent screen includes:

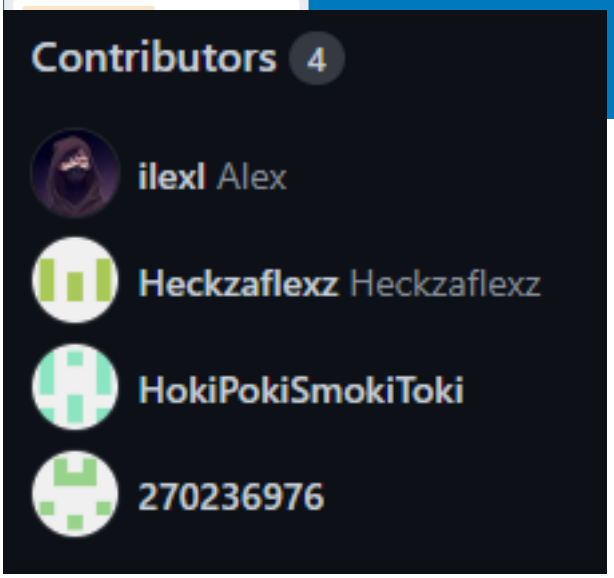
- View School Notice
- View Reports

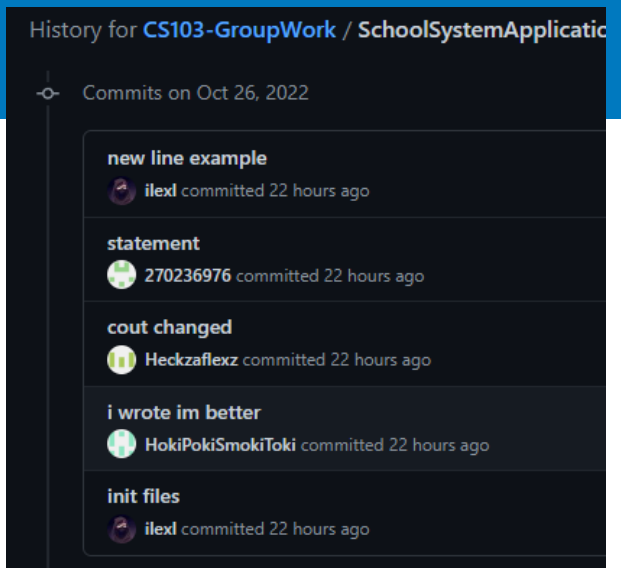
Collaboration Evidence

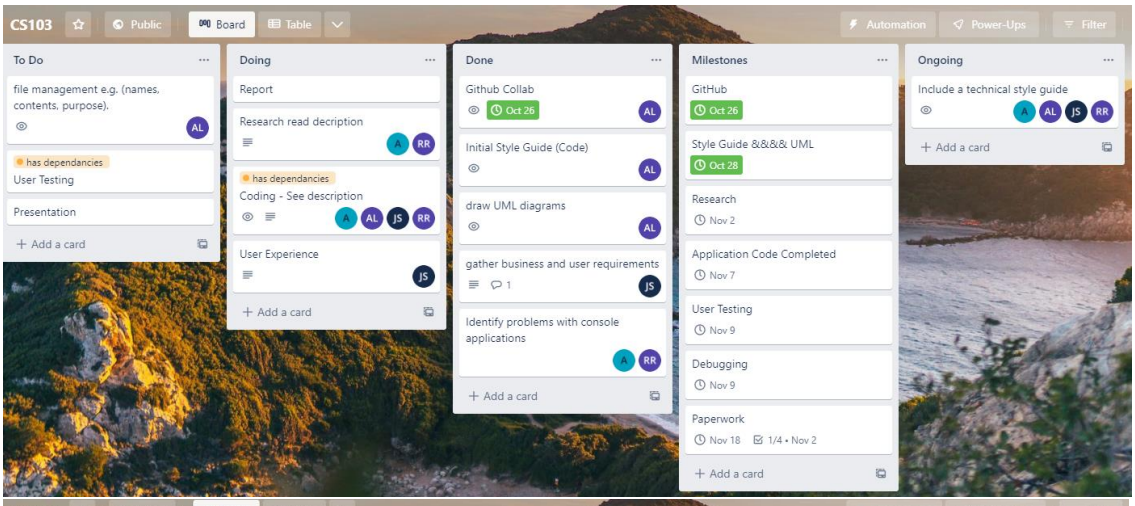
https://trello.com/b/Ycfw7hEi/cs103

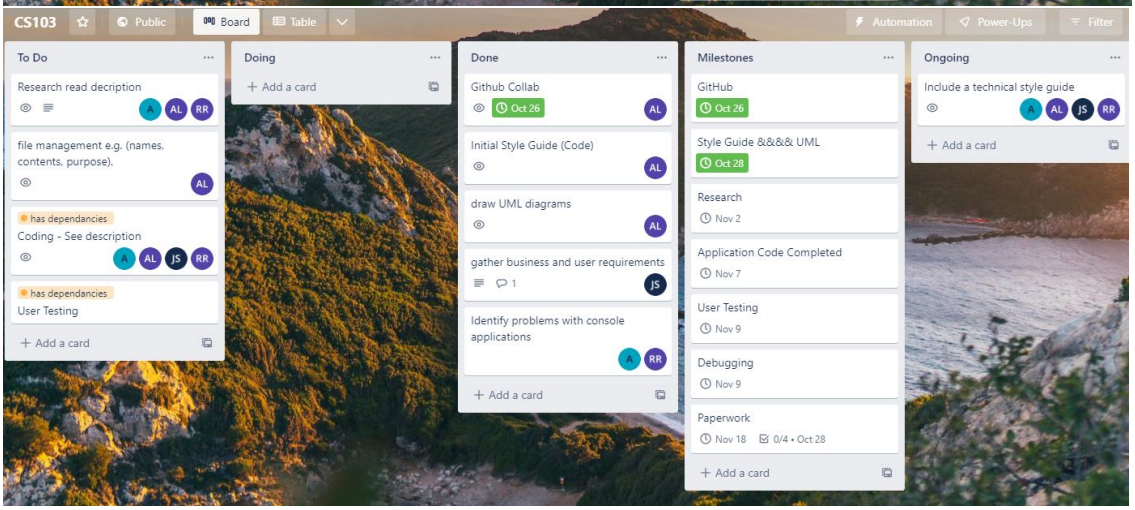
https://github.com/ilexl/CS103-GroupWork

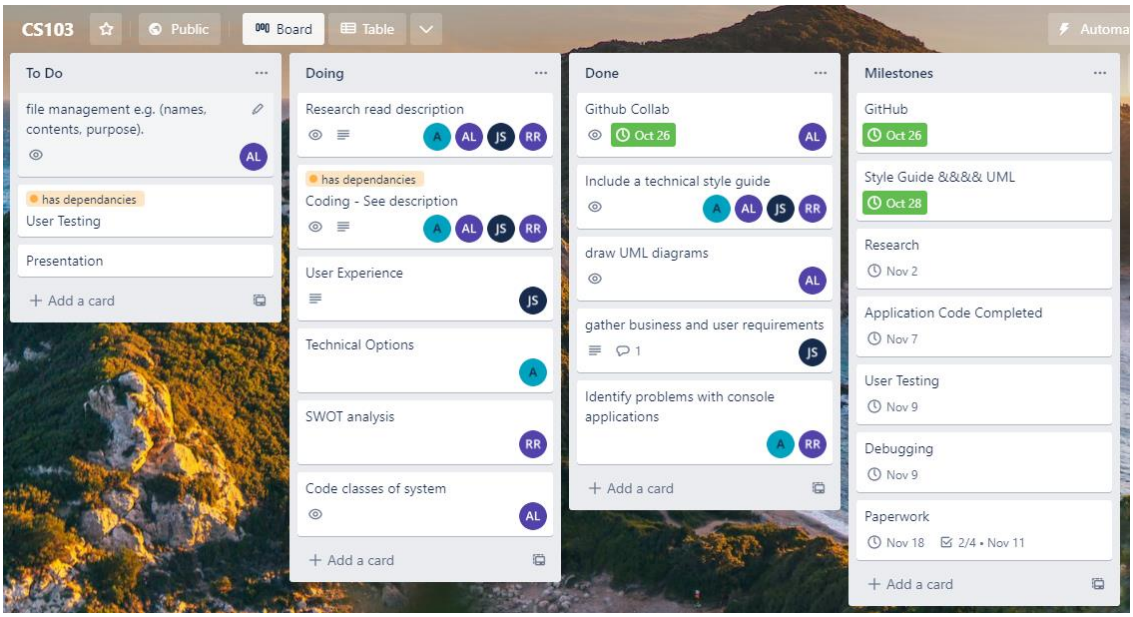


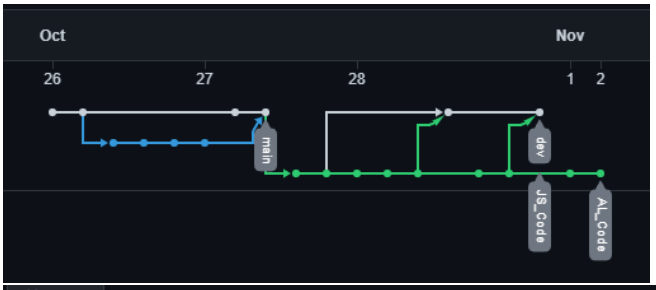


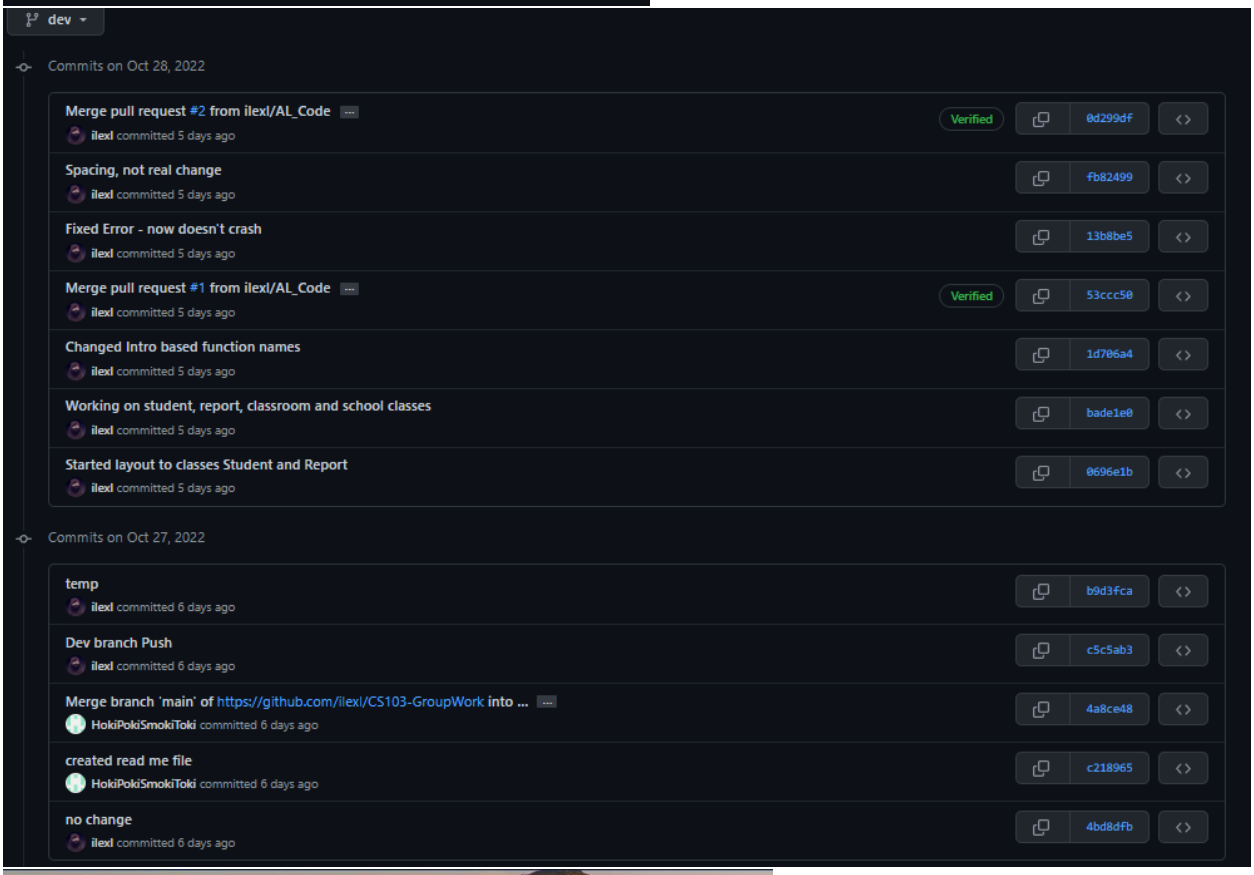


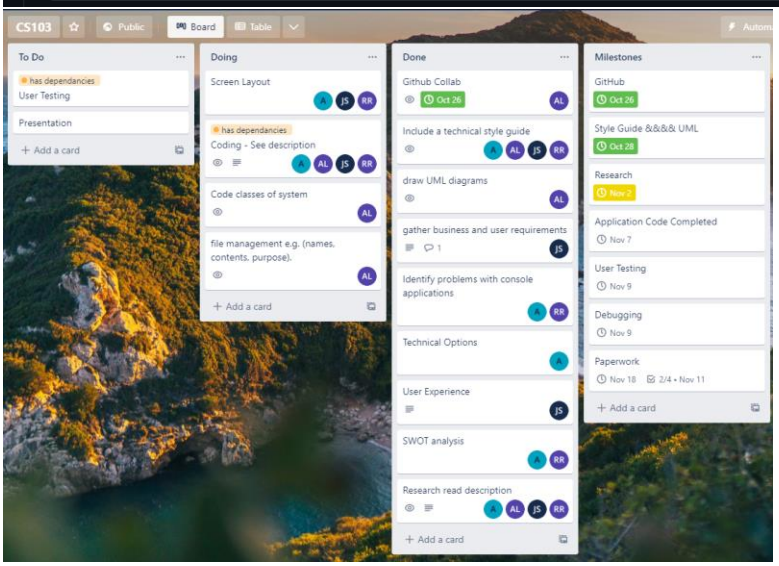

CS103
Public
Board
Table
Automation
Power-Ups
Filter
To Do
file management e.g. (names, contents, purpose).
has dependancies
User Testing
Presentation
Add a card
Doing
Report
Research read decription
has dependancies
Coding - See description
User Experience
Add a card
Done
Github Collab
Oct 26
Initial Style Guide (Code)
draw UML diagrams
gather business and user requirements
Identify problems with console applications
Add a card
Milestones
GitHub
Oct 26
Style Guide &&&& UML
Oct 28
Research
Nov 2
Application Code Completed
Nov 7
User Testing
Nov 9
Debugging
Nov 9
Paperwork
Nov 18
1/4 • Nov 2
Add a card
Ongoing
Include a technical style guide
Add a card

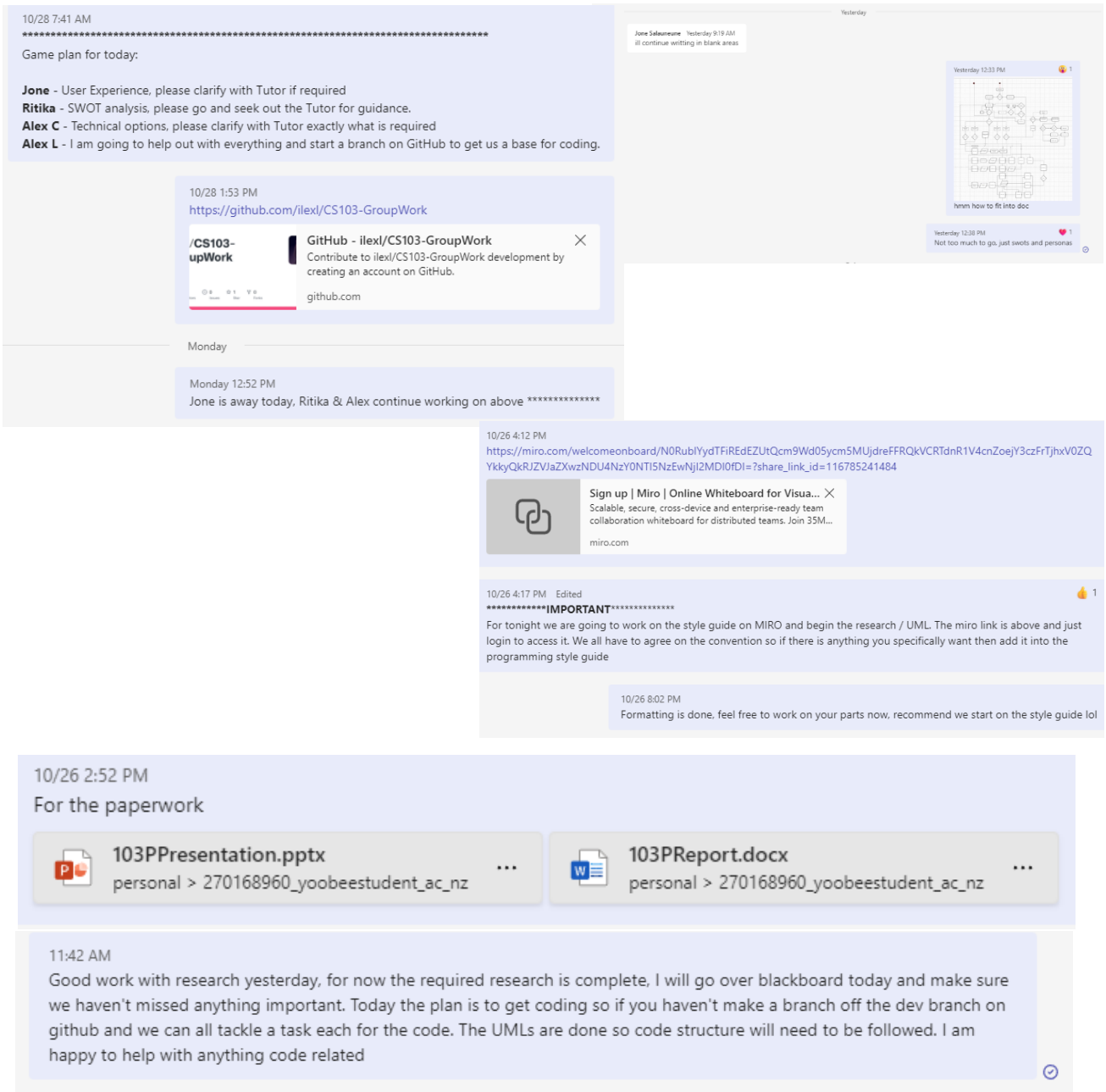

CS103
Public
Board
Table
Automation
Power-Ups
Filter
To Do
Research read decription
file management e.g. (names, contents, purpose).
has dependancies
Coding - See description
has dependancies
User Testing
Add a card
Doing
Add a card
Done
Github Collab
Oct 26
Initial Style Guide (Code)
draw UML diagrams
gather business and user requirements
Identify problems with console applications
Add a card
Milestones
GitHub
Oct 26
Style Guide &&&& UML
Oct 28
Research
Nov 2
Application Code Completed
Nov 7
User Testing
Nov 9
Debugging
Nov 9
Paperwork
Nov 18
0/4 • Oct 28
Add a card
Ongoing
Include a technical style guide
Add a card

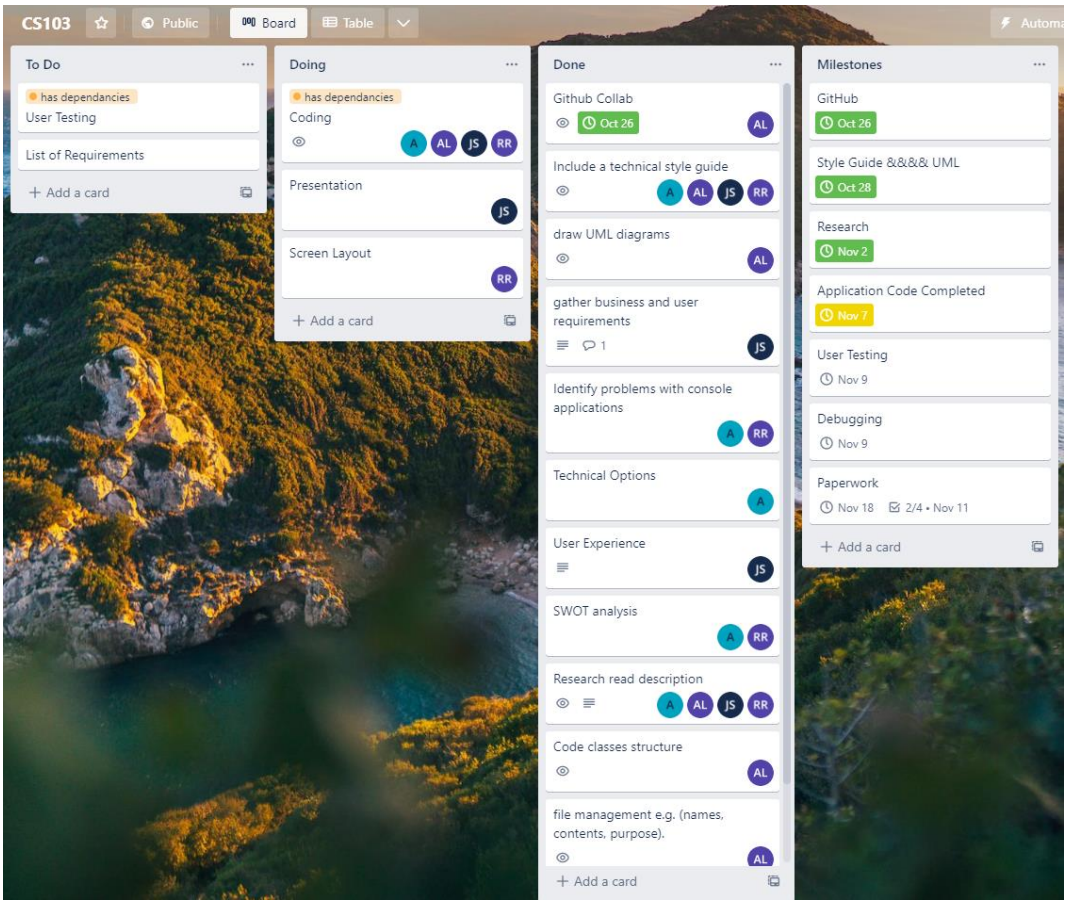

CS103
Public
Board
Table
To Do
file management e.g. (names, contents, purpose).
has dependancies
User Testing
Presentation
Add a card
Doing
Research read description
has dependancies
Coding - See description
User Experience
Technical Options
SWOT analysis
Code classes of system
Add a card
Done
Github Collab
Oct 26
Include a technical style guide
draw UML diagrams
gather business and user requirements
Identify problems with console applications
Add a card
Milestones
GitHub
Oct 26
Style Guide &&&& UML
Oct 28
Research
Nov 2
Application Code Completed
Nov 7
User Testing
Nov 9
Debugging
Nov 9
Paperwork
Nov 18
2/4 • Nov 11
Add a card

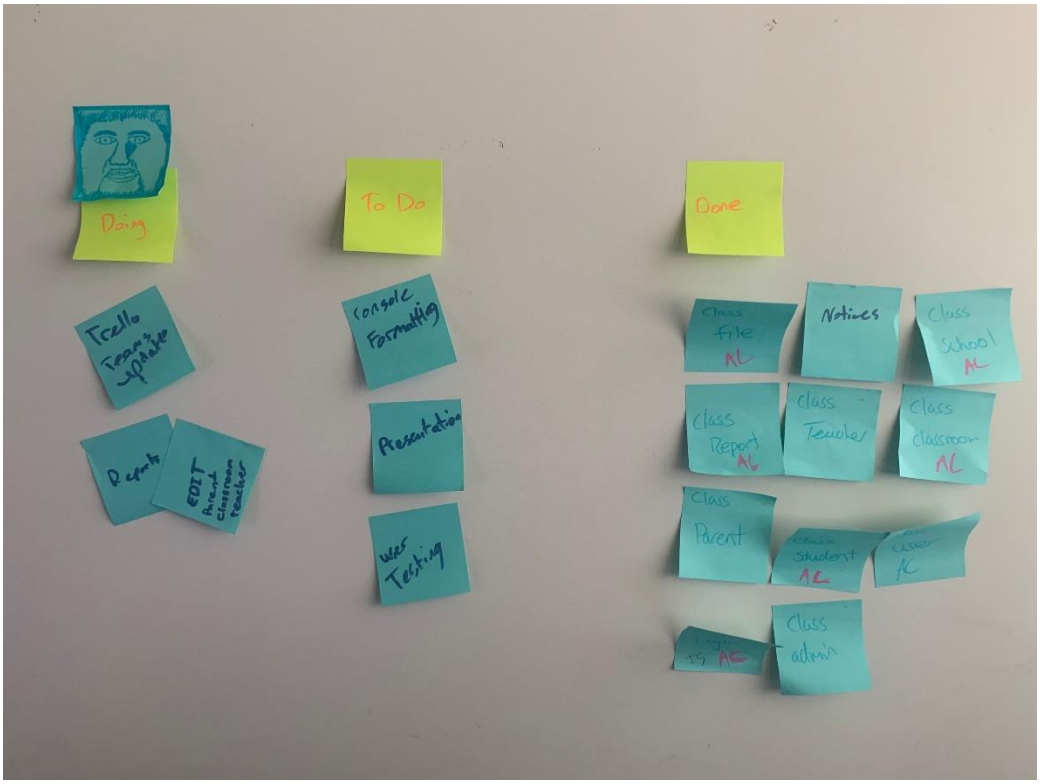

Oct
Nov
26
27
28
1
2
main
dev
JS_Code
AL_Code


dev
Commits on Oct 28, 2022
Merge pull request #2 from ilexl/AL_Code
ilexl committed 5 days ago
Verified
0d299df
Spacing, not real change
ilexl committed 5 days ago
fb82499
Fixed Error - now doesn't crash
ilexl committed 5 days ago
13b8be5
Merge pull request #1 from ilexl/AL_Code
ilexl committed 5 days ago
Verified
53ccc50
Changed Intro based function names
ilexl committed 5 days ago
1d706a4
Working on student, report, classroom and school classes
ilexl committed 5 days ago
bade1e0
Started layout to classes Student and Report
ilexl committed 5 days ago
0696e1b
Commits on Oct 27, 2022
temp
ilexl committed 6 days ago
b9d3fca
Dev branch Push
ilexl committed 6 days ago
c5c5ab3
Merge branch 'main' of https://github.com/ilexl/CS103-GroupWork into ...
HokiPokiSmokiToki committed 6 days ago
4a8ce48
created read me file
HokiPokiSmokiToki committed 6 days ago
c218965
no change
ilexl committed 6 days ago
4bd8dfb


CS103
Public
Board
Table
To Do
has dependancies
User Testing
Presentation
Add a card
Doing
Screen Layout
has dependancies
Coding - See description
Code classes of system
file management e.g. (names, contents, purpose).
Add a card
Done
Github Collab
Oct 26
Include a technical style guide
draw UML diagrams
gather business and user requirements
Identify problems with console applications
Technical Options
User Experience
SWOT analysis
Research read description
Add a card
Milestones
GitHub
Oct 26
Style Guide &&&&& UML
Oct 28
Research
Nov 7
Application Code Completed
Nov 7
User Testing
Nov 9
Debugging
Nov 9
Paperwork
Nov 18
2/4 • Nov 11
Add a card


10/28 7:41 AM
****
Game plan for today:
Jone - User Experience, please clarify with Tutor if required
Ritika - SWOT analysis, please go and seek out the Tutor for guidance.
Alex C - Technical options, please clarify with Tutor exactly what is required
Alex L - I am going to help out with everything and start a branch on GitHub to get us a base for coding.
10/28 1:53 PM
https://github.com/ilexl/CS103-GroupWork
GitHub - ilexl/CS103-GroupWork
Contribute to ilexl/CS103-GroupWork development by creating an account on GitHub.
github.com
Monday
Monday 12:52 PM
Jone is away today, Ritika & Alex continue working on above **********
Yesterday
Jone Salaunerre Yesterday 9:19 AM
ill continue writting in blank areas
Yesterday 12:33 PM
hmm how to fit into doc
Yesterday 12:38 PM
Not too much to go, just swots and personas
10/26 4:12 PM
https://miro.com/welcomeonboard/N0RublYydTFiREdEZUtQcm9Wd05ycm5MUjdreFFRQkVCRTdnR1V4cnZoejY3czFrTjhxV0ZQYkkyQkRJZVJaZXwzNDU4NzY0NTI5NzEwNjI2MDI0fDI=?share_link_id=116785241484
Sign up | Miro | Online Whiteboard for Visua...
Scalable, secure, cross-device and enterprise-ready team collaboration whiteboard for distributed teams. Join 35M...
miro.com
10/26 4:17 PM Edited
**********IMPORTANT**********
For tonight we are going to work on the style guide on MIRO and begin the research / UML. The miro link is above and just login to access it. We all have to agree on the convention so if there is anything you specifically want then add it into the programming style guide
10/26 8:02 PM
Formatting is done, feel free to work on your parts now, recommend we start on the style guide lol
10/26 2:52 PM
For the paperwork
103PPresentation.pptx
personal > 270168960_yoobeestudent_ac_nz
103PReport.docx
personal > 270168960_yoobeestudent_ac_nz
11:42 AM
Good work with research yesterday, for now the required research is complete, I will go over blackboard today and make sure we haven't missed anything important. Today the plan is to get coding so if you haven't make a branch off the dev branch on github and we can all tackle a task each for the code. The UMLs are done so code structure will need to be followed. I am happy to help with anything code related


CS103
Public
Board
Table
To Do
has dependancies
User Testing
List of Requirements
Add a card
Doing
has dependancies
Coding
Presentation
Screen Layout
Add a card
Done
Github Collab
Oct 26
Include a technical style guide
draw UML diagrams
gather business and user requirements
Identify problems with console applications
Technical Options
User Experience
SWOT analysis
Research read description
Code classes structure
file management e.g. (names, contents, purpose).
Add a card
Milestones
GitHub
Oct 26
Style Guide &&&& UML
Oct 28
Research
Nov 2
Application Code Completed
Nov 7
User Testing
Nov 9
Debugging
Nov 9
Paperwork
Nov 18
2/4 • Nov 11
Add a card


Doing
To Do
Done
Trello Team Update
Console Formatting
Presentation
User Testing
EDIT
Natives
Class School
Class Report
Class Teacher
Class Classroom
Class Parent
Class admin

Alex L created presentation and report

Oct 26

For tonight we are going to work on the style guide on MIRO and begin the research / UML. The miro link is above and just login to access it. We all have to agree on the convention so if there is anything you specifically want then add it into the programming style guide

Oct 26

Game plan for today: Jone - User Experience, please clarify with Tutor if required Ritika - SWOT analysis, please go and seek out the Tutor for guidance. Alex C - Technical options, please clarify with Tutor exactly what is required Alex L - I am going to help out with everything and start a branch on GitHub to get us a base for coding.

Oct 28

Alex L - finished UML Diagrams

Nov 1

Research - Alex, Jone, Ritika - Working on Swots and Personas

Nov 1

Research Complete

Nov 2

OPTION 1 - Jone - student->options - show options - get input for options - take appriate action parent->options - show options - get input for options - take appropriate action

Nov 7

OPTION 2 - Ritika teacher->options - show options - get input for options - take appropriate action

Nov 7

OPTION 3 Alex.L admin->options - edit student details - edit parent details - add new parent->get more info and store in parent file appropriately - delete parent -> delete parent from parent file - edit classroom - show list of parents - show list of parents - school details - show list of teachers - new option->edit logins

Nov 7

OPTION 4 - Alex.C main -> switch - change to login && &&Sign up---- not just login

Nov 7

Code Update - Teacher->options - show options - get input for options - take appropriate action - edit parent - edit classroom - edit teacher - reports. That's all we have left to do :)

Nov 8

Alex L created the console formatting

Group Update - WE have finished coding the application, however WE need to test it and format the console outputs to make it pretty this is OUR task tomorrow (:

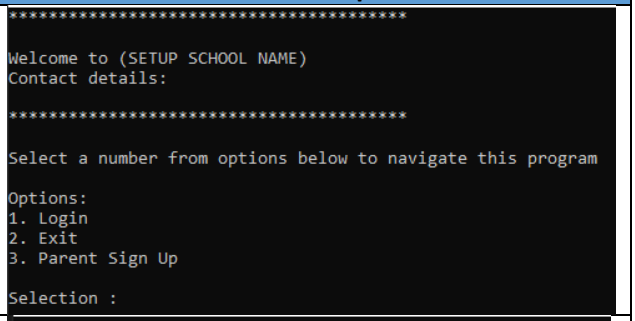
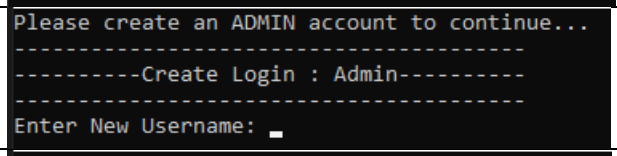
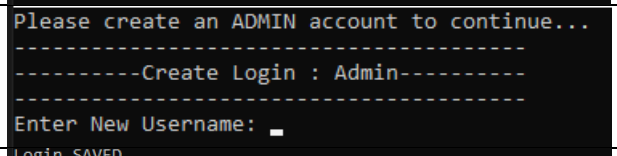
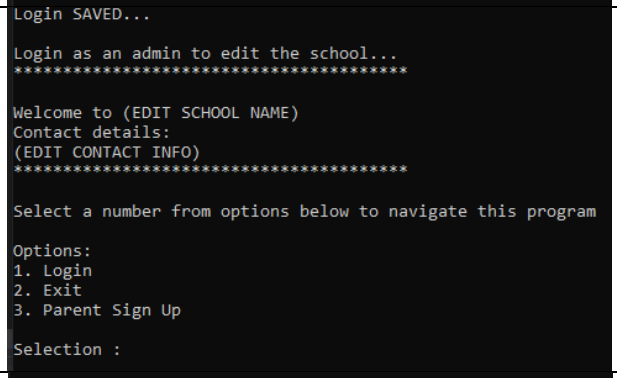
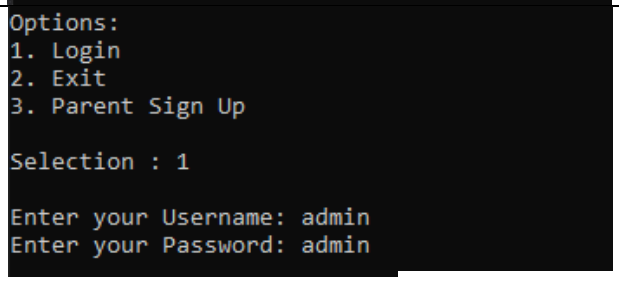
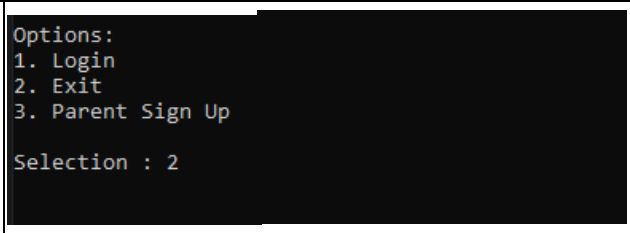
Nov 9

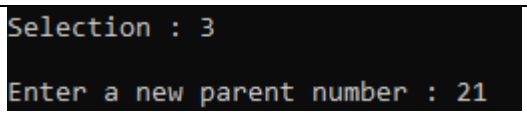
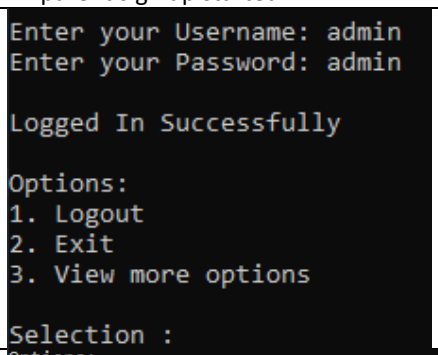
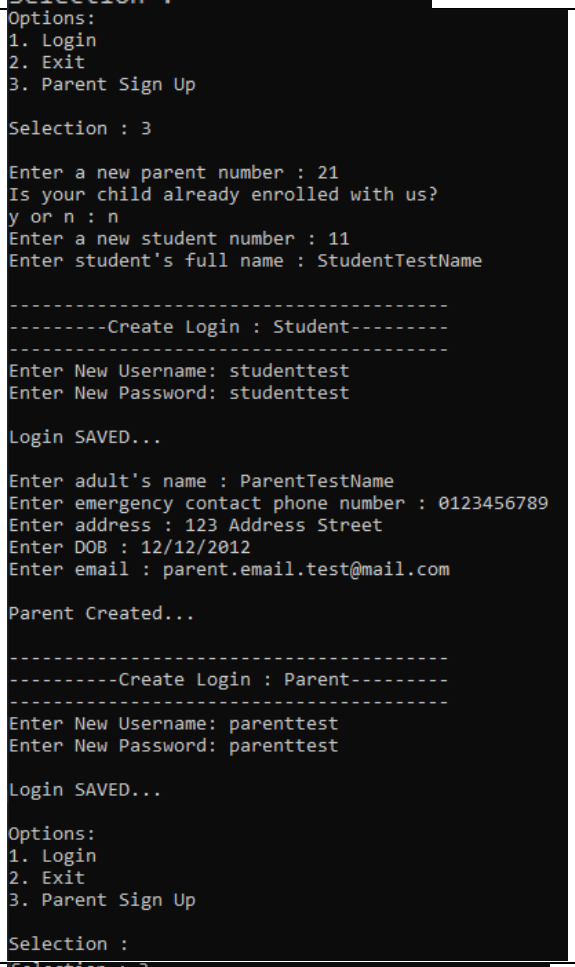
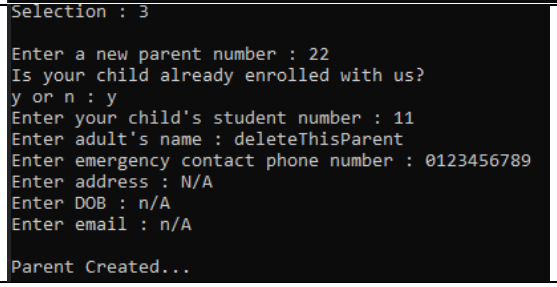
User Testing Feedback, need to update Parent sign ups Example cout << "Enter an 8 digit number"

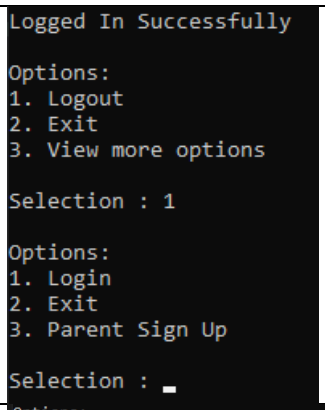
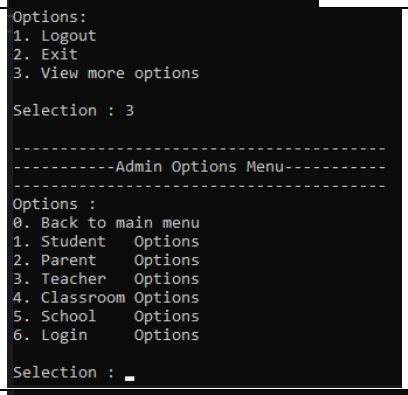
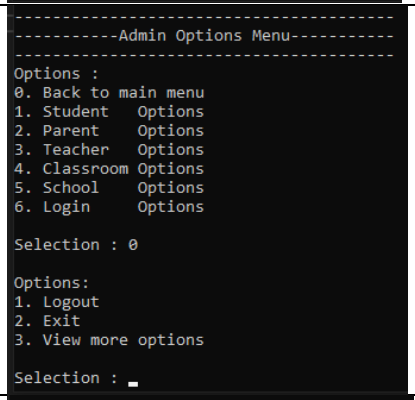
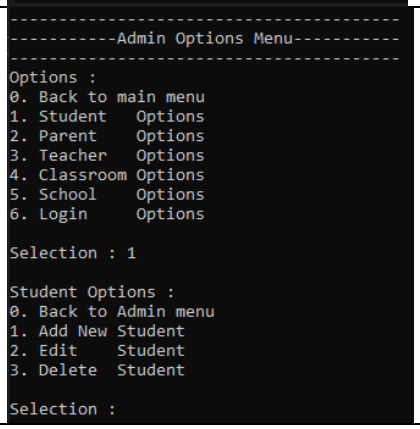
Nov 10

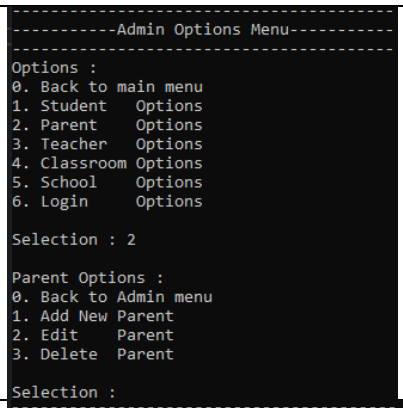
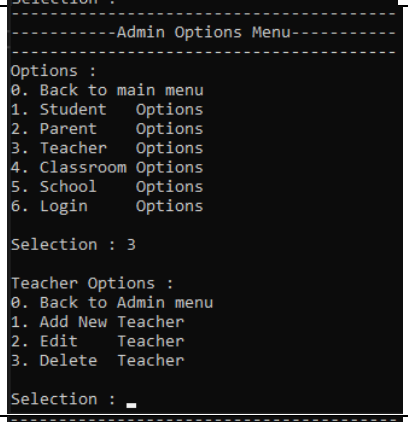
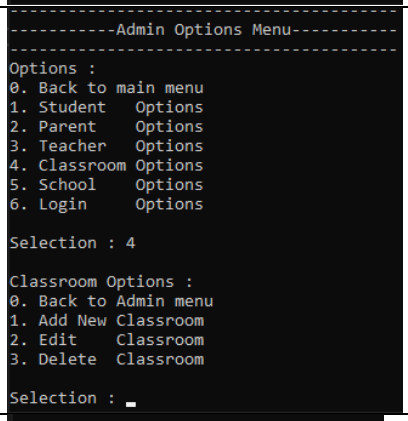
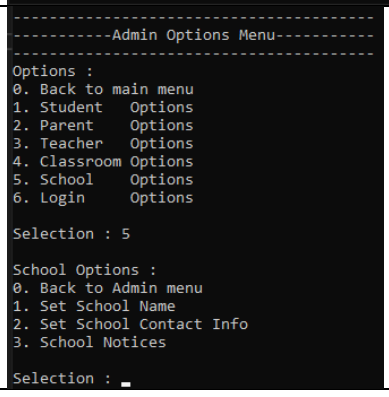
Testing

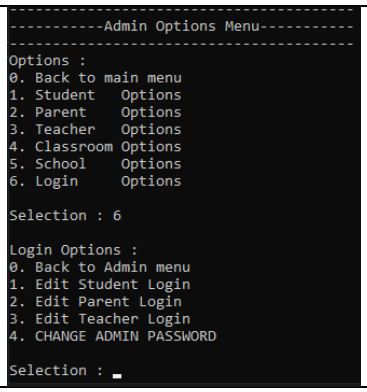
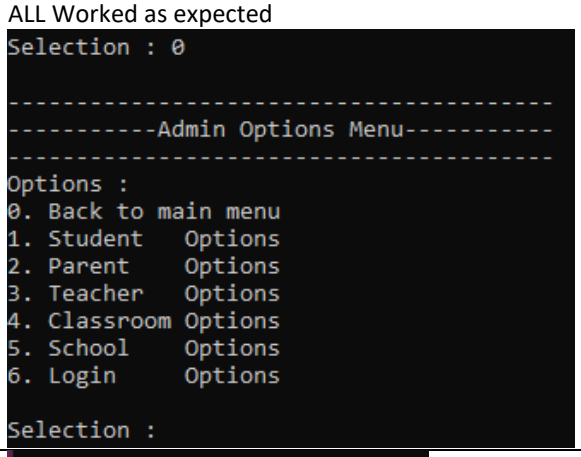
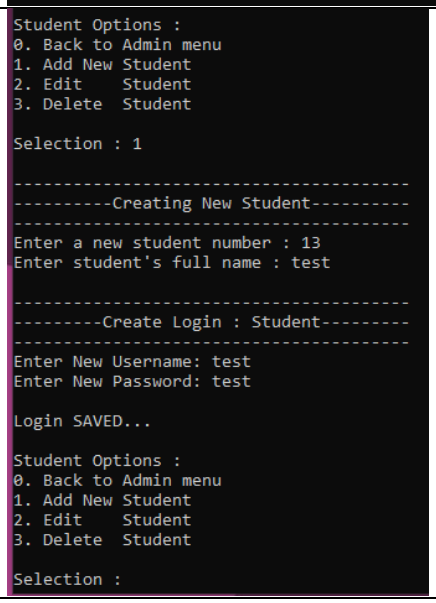
Valid Input Testing

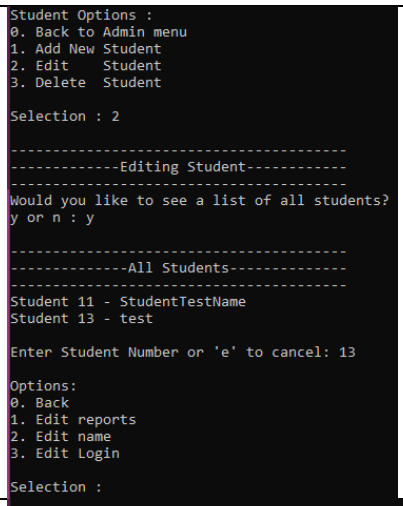
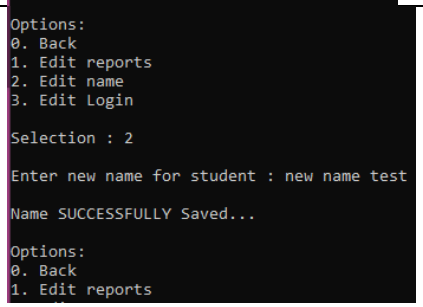
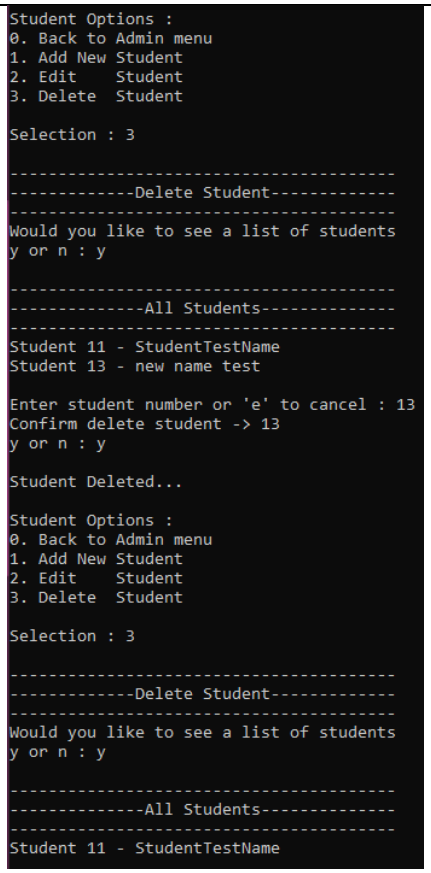
Input	Expected Output	Actual Output
Opened the program (Previously Setup)	Welcome user and prompt to select from main menu	
Opened the program (Not Setup Before)	Create ADMIN account and inform them to setup the school.	
Opened the program (Not Setup PROPERLY Before)	Create ADMIN account and inform them to setup the school.	
Setup ADMIN account on launch	Prompts admin to login to edit the school once signed up and displays welcome screen as normal.	
Selection Option 1 (Login) in Main Menu	Prompts a login	
Selection Option 2 (Exit) in Main Menu	Program should exit	 ^ Program exited

Selection Option 3 (Sign Up) in Main Menu	Start signing up a parent	 ^^ parent sign up started
Login in with correct details		
Enter valid parent sign up details	Sign the parent and their student up to the system	
Select student already in system for parent sign up (valid)	Use the student's name already in system and attach student to the parent	

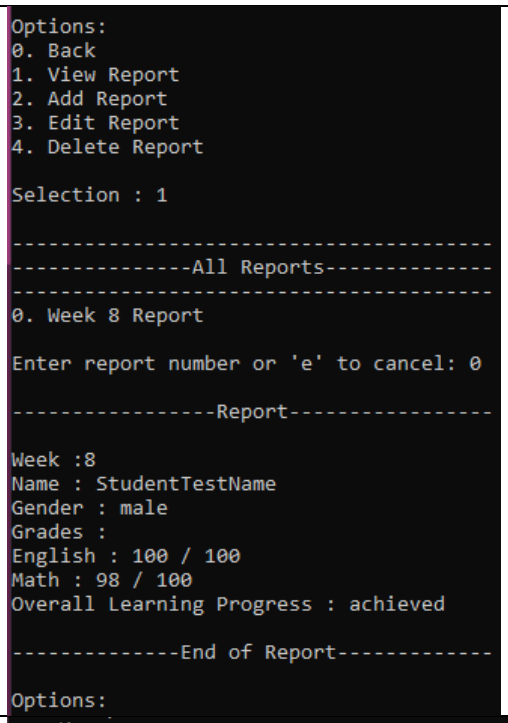
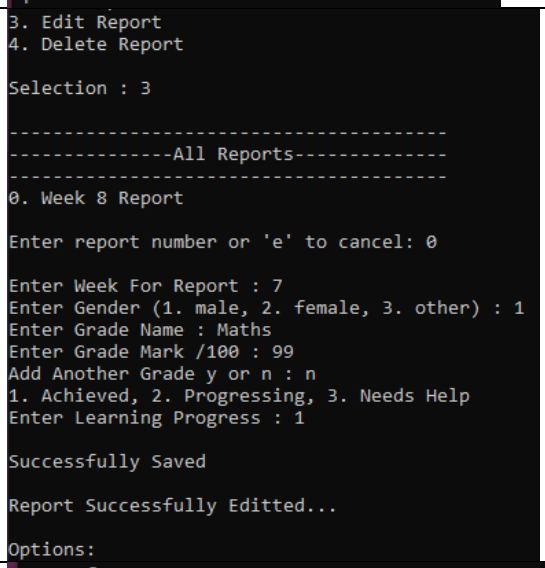
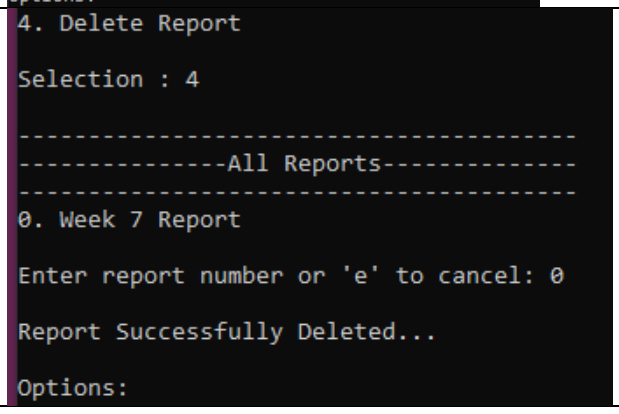
(Logged in as admin) Main menu option 1 (Logout)	Logout and display main menu	
(Logged in as admin) Main menu option 3 (View more options)	Show sub menu admin	
(Logged in as admin) Admin Options Menu 0 (back to main menu)	Go back to main menu	
(Logged in as admin) Admin Options Menu 1 (student menu)	Go to sub admin menu to manage students	

(Logged in as admin) Admin Options Menu 2 (parent menu)	Go to sub admin menu to manage parents		
(Logged in as admin) Admin Options Menu 3 (teacher menu)	Go to sub admin menu to manage teachers		
(Logged in as admin) Admin Options Menu 4 (classroom menu)	Go to sub admin menu to manage classrooms		
(Logged in as admin) Admin Options Menu 5 (school menu)	Go to sub admin menu to manage school information		

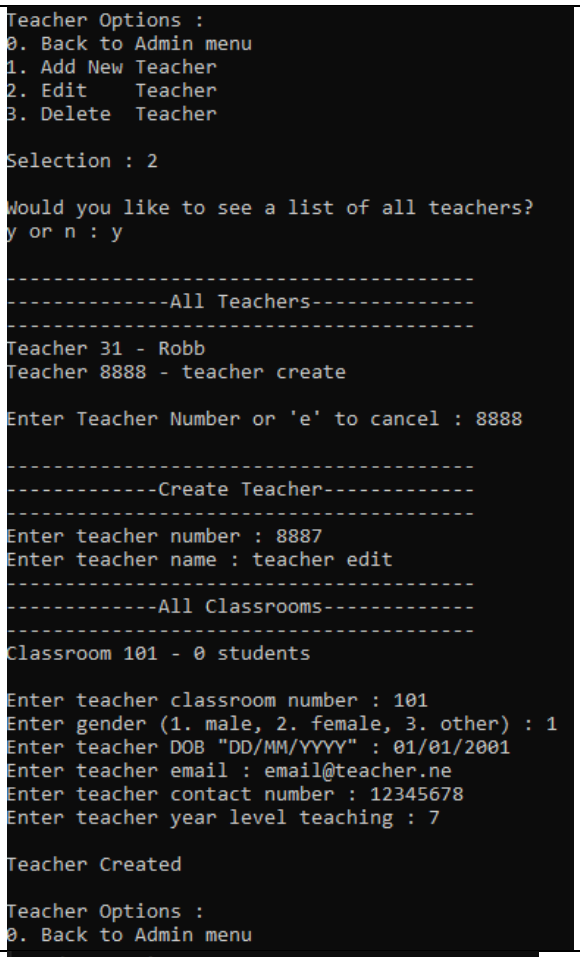
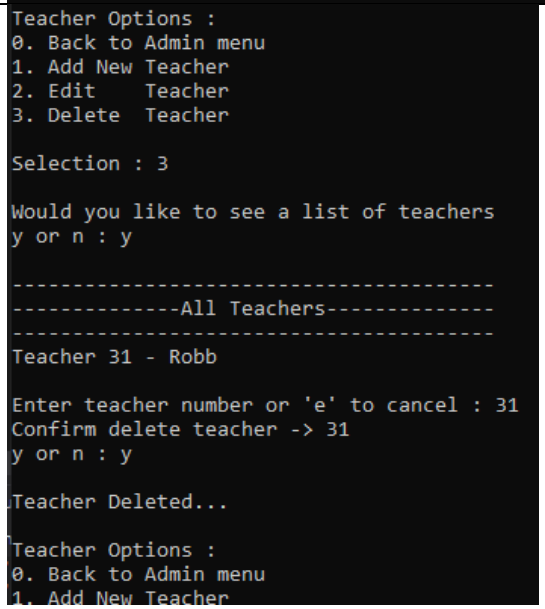
(Logged in as admin) Admin Options Menu 6 (login menu)	Go to sub admin menu to manage logins	
(Logged in as admin) Admin Options SUB Menu 0 (Back to admin menu)	Go back to admin menu options	ALL Worked as expected 
(Logged in as admin) Create a student	Create a student who is stored in memory and to the csv on completion	

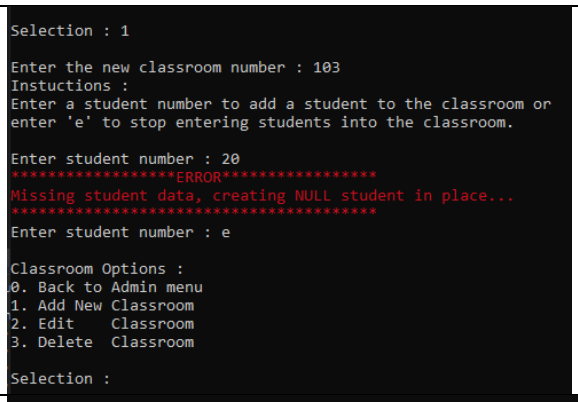
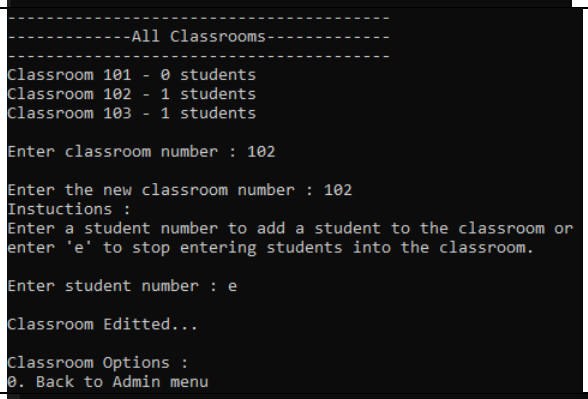
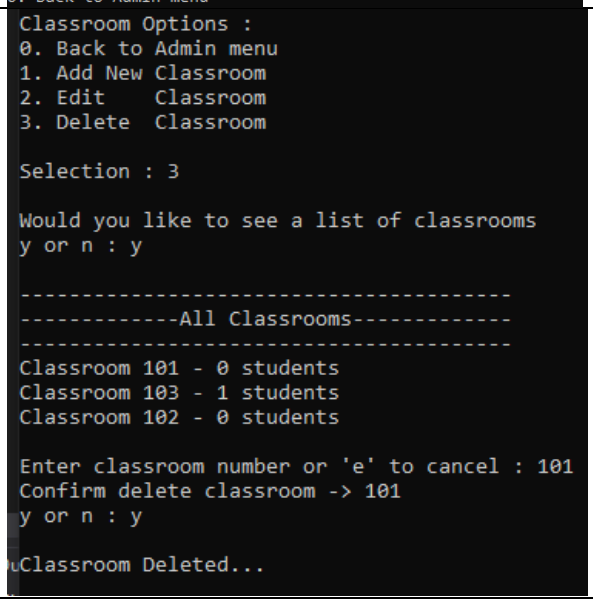
(Logged in as admin) Edit a student	Open Edit menu of a student	
(Logged in as admin) Student selected to edit Edit name of student	Edit the name of the student in memory and to the csv	
(Logged in as admin) Delete a student	Delete a student from memory and in the csv	

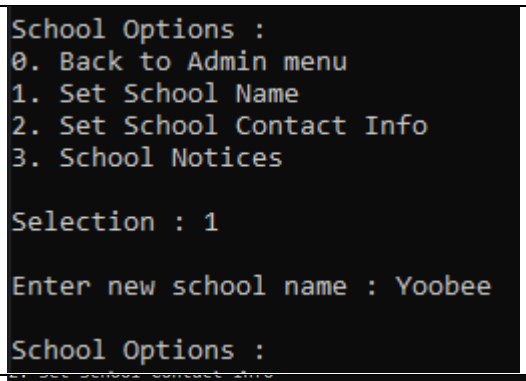
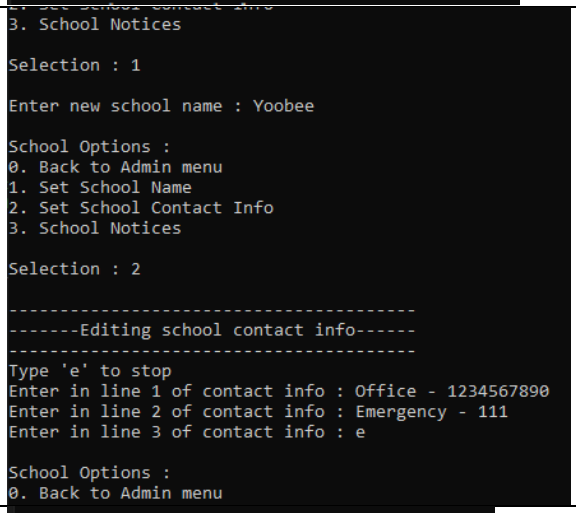
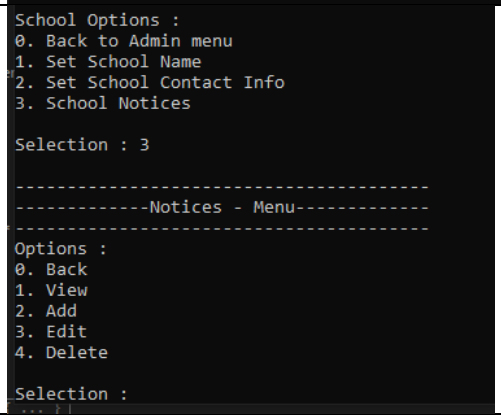
(Logged in as admin) Report option edit student	Open Report sub menu	Student 11 - StudentTestName Enter Student Number or 'e' to cancel: 11 Options: 0. Back 1. Edit reports 2. Edit name 3. Edit Login Selection : 1 Options: 0. Back 1. View Report 2. Add Report 3. Edit Report 4. Delete Report
(Logged in as admin) Report option edit student Add Report	Create a student report and store in memory and csv	Options: 0. Back 1. View Report 2. Add Report 3. Edit Report 4. Delete Report Selection : 2 Enter Week For Report : 8 Enter Gender (1. male, 2. female, 3. other) : 1 Enter Grade Name : English Enter Grade Mark /100 : 100 Add Another Grade y or n : y Enter Grade Name : Math Enter Grade Mark /100 : 98 Add Another Grade y or n : n 1. Achieved, 2. Progressing, 3. Needs Help Enter Learning Progress : 1 Successfully Saved Options: 0. Back 1. View Report 2. Add Report 3. Edit Report 4. Delete Report Selection :

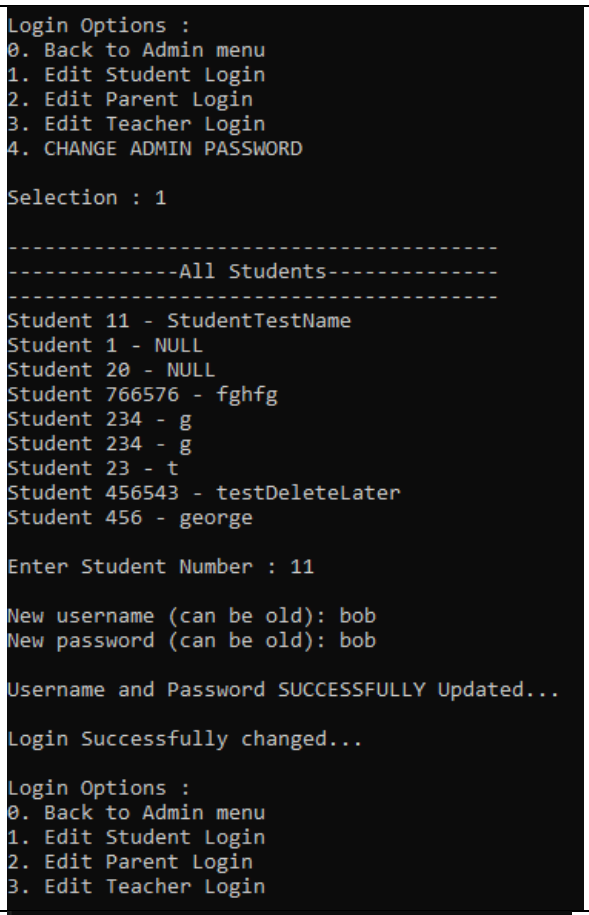
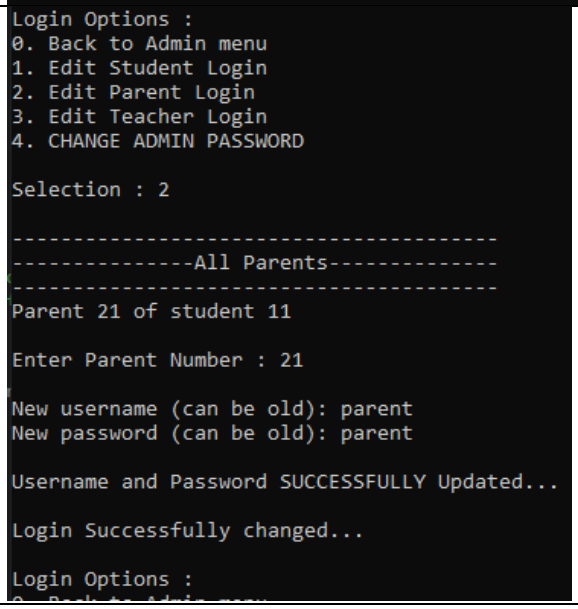
View Report	View a list of reports -> view a specific report and what is in it.	
Edit Report	Edits a report overwriting the one selected and stored in memory and csv	
Delete Report	Deletes a reports from memory and in the csv	

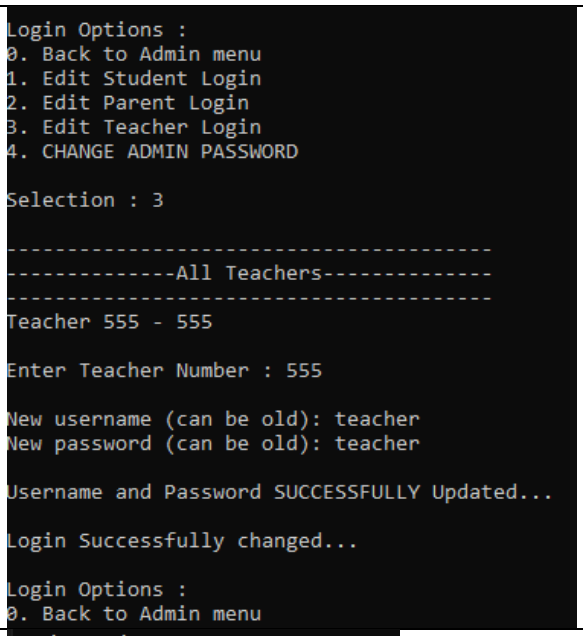
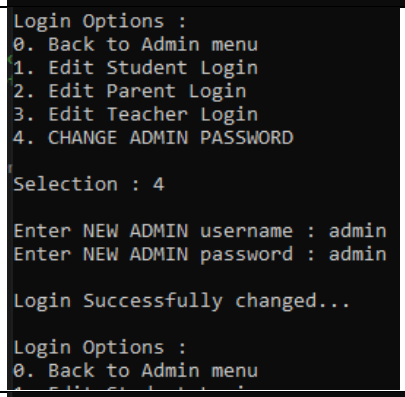
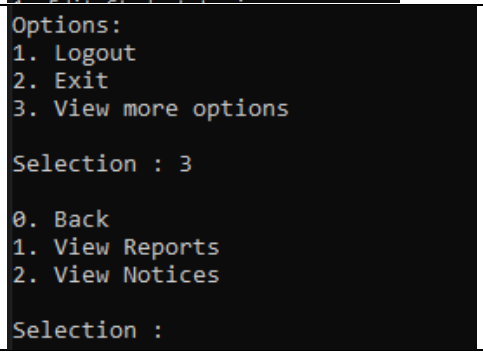
(Logged in as admin) Edit a parent	Edit a parent and overwrite the parent data in memory and csv	Same as in main menu signing up a parent, works the exact same and as expected Parent Options : 0. Back to Admin menu 1. Add New Parent 2. Edit Parent 3. Delete Parent Selection : 1 Enter a new parent number :
(Logged in as admin) Delete a parent	Delete a parent from memory and in the csv	Parent Options : 0. Back to Admin menu 1. Add New Parent 2. Edit Parent 3. Delete Parent Selection : 3 Would you like to see a list of parents y or n : y ---------- ----------All Parents---------- ---------- Parent 21 of student 11 Parent 233 of student 11 Enter parent number or 'e' to cancel : 233 Confirm delete parent -> 233 y or n : y Parent Deleted... Parent Options :
(Logged in as admin) Create a teacher	Create a teacher who is stored in memory and to the csv on completion	Teacher Options : 0. Back to Admin menu 1. Add New Teacher 2. Edit Teacher 3. Delete Teacher Selection : 1 ---------- ----------Create Teacher---------- ---------- Enter teacher number : 8888 Enter teacher name : teacher create ---------- ----------All Classrooms---------- ---------- Classroom 101 - 0 students Enter teacher classroom number : 101 Enter gender (1. male, 2. female, 3. other) : 1 Enter teacher DOB "DD/MM/YYYY" : 01/01/2001 Enter teacher email : email@teacher.com Enter teacher contact number : 123456789 Enter teacher year level teaching : 5 Teacher Created ---------- ---------Create Login : Teacher--------- ---------- Enter New Username: teacher1 Enter New Password: teacher1 Login SAVED... Teacher Options : 0. Back to Admin menu

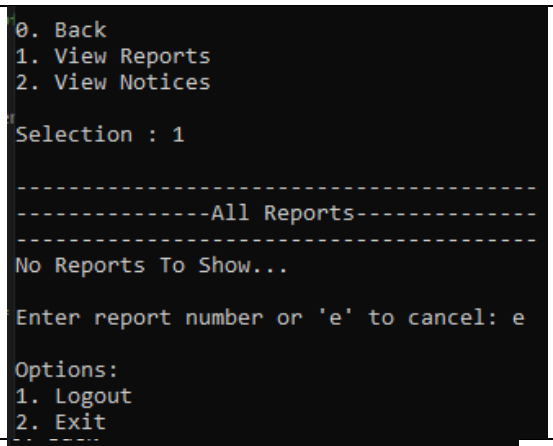
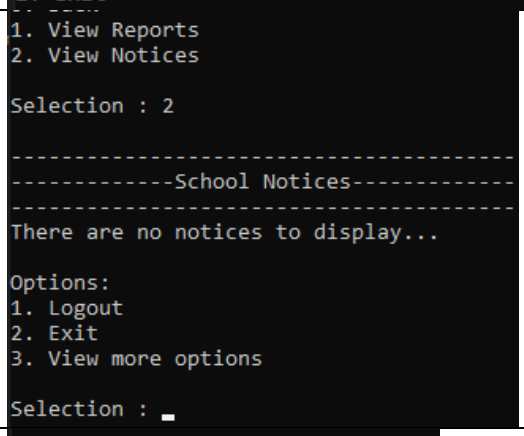
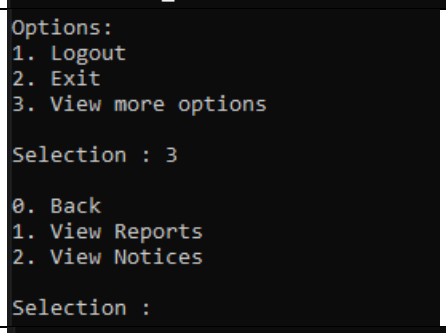
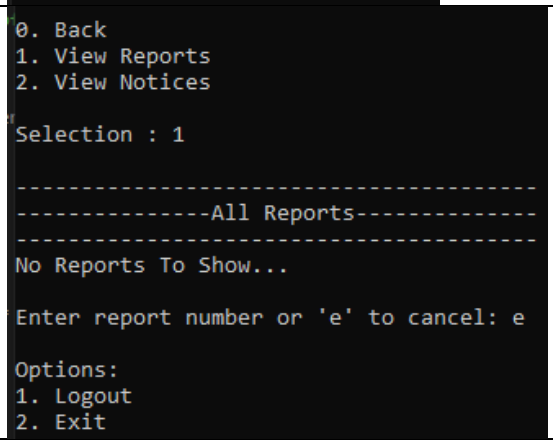
(Logged in as admin) Edit a teacher	Edit a teacher and overwrite data in memory and csv	
(Logged in as admin) Delete a teacher	Delete a teacher from memory and in the csv	

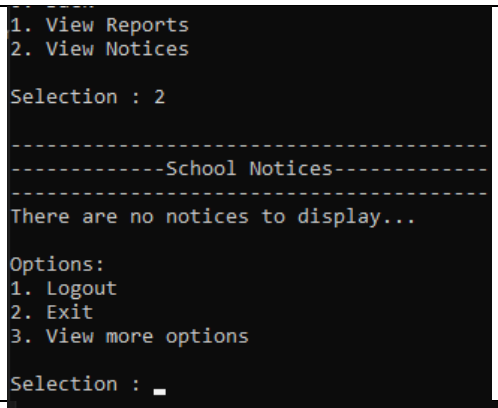
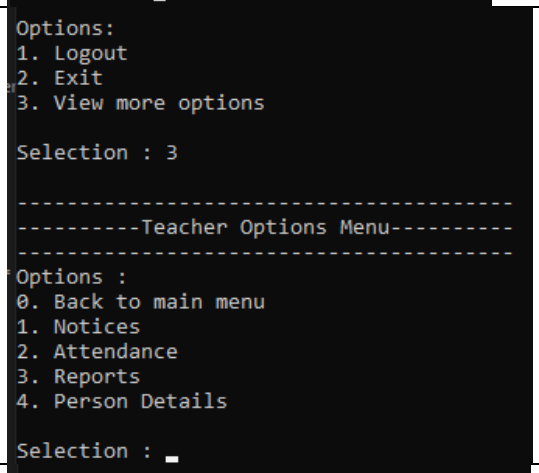
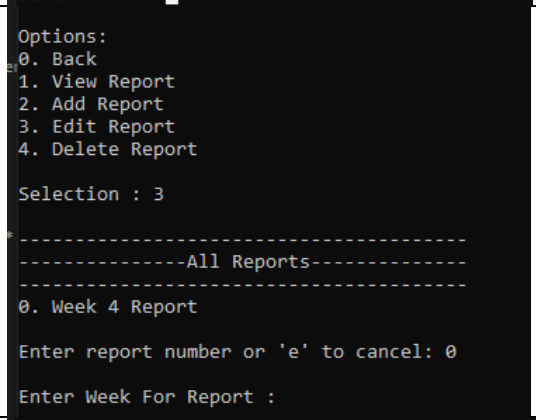
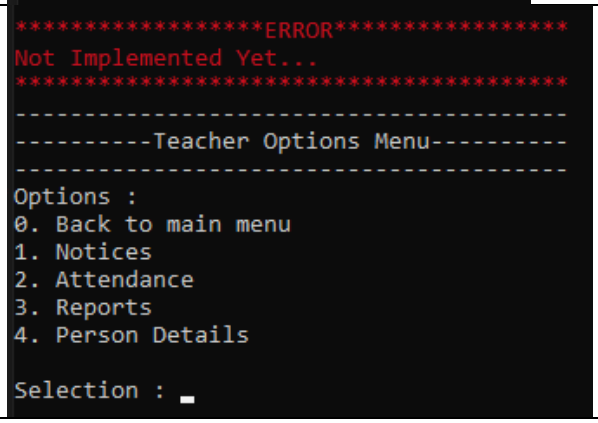
(Logged in as admin) Create a Classroom	Create a classroom who is stored in memory and to the csv on completion	
(Logged in as admin) Edit a Classroom	Open edit menu of a classroom	
(Logged in as admin) Delete a Classroom	Delete a classroom from memory and in the csv	

(Logged in as admin) Set School Name	Sets the school name to be saved to memory and to the csv	
(Logged in as admin) Set School Contact Info	Setup the school contact info and save data to memory and csv file	
(Logged in as admin) School -> Notices	Take to notices menu	

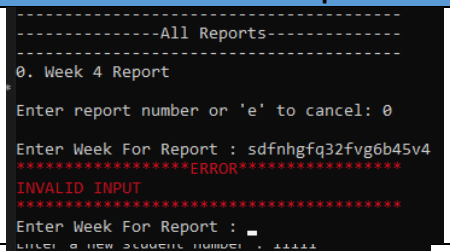
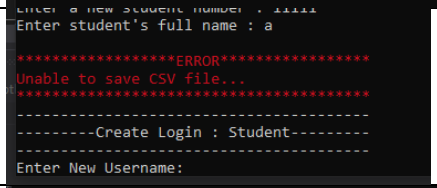
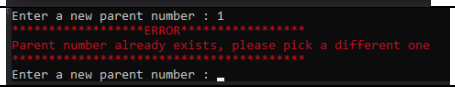
(Logged in as admin) Edit Student Login	Create a student who is stored in memory and to the csv on completion	
(Logged in as admin) Edit Parent Login	Edit a student who is stored in memory and to the csv on completion	

(Logged in as admin) Edit Teacher Login	Delete a student from memory and in the csv	
(Logged in as admin) Edit THE ADMIN Login	Edit the login details in memory and in the csv	
(Logged in as parent) View more options	Show the parent options	

(Logged in as parent) View student's reports	Show students reports	
(Logged in as parent) View school notices	Show school notices	
(Logged in as student) view more options	Show the student options	
(Logged in as student) View own reports	Show its own reports	

(Logged in as student) View school notices		
(Logged in as teacher) Show more options	Show the teacher options	
(Logged in as teacher) Edit student report	Edits / Views the reports of a student	
(Logged in as teacher) Attendance	Display error of not implemented yet	

Invalid Testing

Input	Expected Output	Actual Output
Any Invalid Input of random string with or without spaces	Error to be handled and displayed to user without crashing the program	
CSV File open when using program	Error displaying it can't save to the csv	
Try to create a duplicate of already existing data	Display error and make user try again	

User Testing

User 1: "ADMIN TASKS" entering letter for cancel command not preferable, when adding students to classroom confused by Null student error, adding a student to database does not give confirmation, edit function is tiresome having to reinput all the information instead of selection what you want to edit. User would prefer text instead of numbers for selection.

User 2: "TEACHER TASKS" typed in name of student for input instead of corresponding number, 'grade name' is confusing and vague, report section confused user due to gender and grade name selection.

User 3: "PARENT TASKS" parent number was confusing due duplicates, 'Emergency contact' should be direct and say 'Emergency Contact Number.

User 4: "ALL TASKS" unable to decipher selection process (description added at start of program), continued to select exit program after signing up "accidentally".

User 5: "PARENT TASKS" was unsure if he managed to reach the proper report page, found the letter spacing to be well made and selection process easy to understand.

User Testing Modifications:

- Sign up numbering system updated to an 8-digit number.
- Included information at start of program to help users navigate the selection process.
- Fixed recursion error when adding a student.
- Added confirmation when adding students to a classroom.
- Changed text for error from "NULL" to "empty" when selection doesn't exist.

Due to time constraints, we are unable to change the editing process from re-inputting all data to selecting which data to edit. If asked to create this program again we would ask if the client would like to edit specific inputs.

References

Admin, E. S. (2017, January 2). *Requirements of School Management System Software*. Edsys. https://www.edsys.in/requirements-of-school-management-system-software/

Cheng, V. (2018, July 7). *Designing a Student Information System: An UX Case Study*. Prototypr. https://blog.prototypr.io/designing-a-student-attendance-and-project-tracking-system-an-ux-case-study-97378795a545

ClickUp<sup>TM | One app to replace them all. (n.d.). ClickUp. <span>https://clickup.com/

Console Applications - RAD Studio. (n.d.). https://docwiki.embarcadero.com/RADStudio/Sydney/en/Console_Applications

Cortuk, D. (2021, December 12). *Version Control Software Comparison: Git, Mercurial,CVS, SVN*. Medium. https://medium.com/@derya.cortuk/version-control-software-comparison-git-mercurial-cvs-svn-21b2a71226e4

Goswami, G. (2022, September 26). *7 Reasons Why Schools Need School Information System*. Techjockey.com Blog. https://www.techjockey.com/blog/school-information-system

Gradelink Reviews & Product Details. (2019). Gradelink Reviews & Product Details. https://www.g2.com/products/gradelink/reviews

Manage Your Teams Projects From Anywhere | Trello. (n.d.). https://trello.com/

Maze. (2022, September 8). *11 Brilliant UX Design Examples to Learn From in 2022*. https://maze.co/collections/ux-ui-design/ux-examples/

Microsoft. (n.d.). *Microsoft – Cloud, computers, apps en games*. https://www.microsoft.com/nl-nl/

PowerSchool SIS Reviews & Product Details. (2019). PowerSchool SIS Reviews & Product Details. https://www.g2.com/products/powerschool-sis/reviews#survey-response-4650284

Student Information System: Investigating User Experience (UX). (2021, January). researchgate.net. https://www.researchgate.net/publication/349742387_Student_Information_System_Investigating_User_Experience_UX

What are the limits of console applicati - C++ Forum. (n.d.). https://cplusplus.com/forum/beginner/168916/

What is SIS? Purpose of Student Information System. (2021, September 14). Fedena Blog. https://fedena.com/blog/2021/09/what-is-sis-purpose-of-student-information-system.html