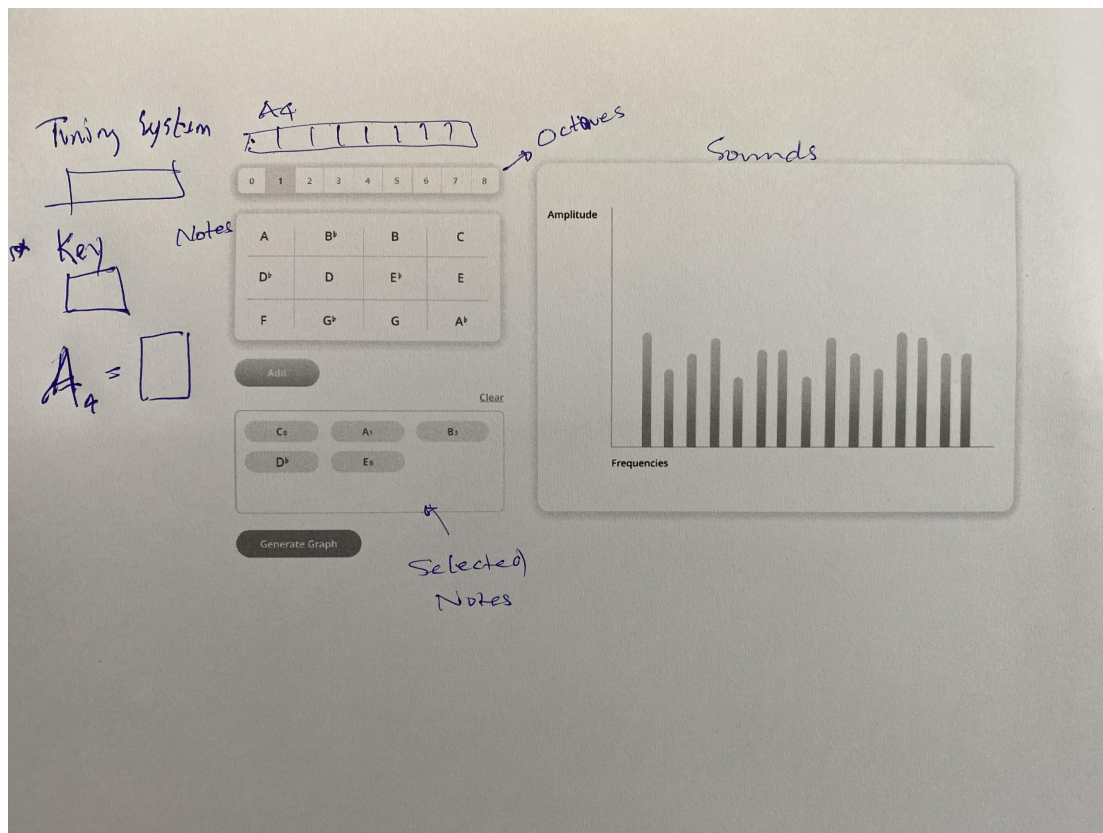


Criterion B: DESIGN

Prototype and Annotation

As it is not possible to meet face to face with the client during the Covid-19 pandemic, I researched prototyping tools that can be used to share the prototype with the client. I opted to use Adobe XD as my prototyping tool. After designing the first prototype, a link to the interactive prototype was shared with the client where he may also provide feedback in his own time. We have had several discussions regarding the prototype. Below are some interactions.

Initial prototype during discussion with client



Subsequent prototype and online interaction

Tuning for A4

432 434 436 438 440 442 444 446

View Chords Logout

Octave

0 1 2 3 4 5 6 7 8

Note

A	B ^b	B	C
D ^b	D	E ^b	E
F	G ^b	G	A ^b

Add

Selected Notes in Chord Clear

Generate Graph 1

Cancel Submit

Clement Guest 11 minutes ago

is this where the graph is?

Reply

Clement Guest 15 minutes ago

by graph does this mean harmonic series?

Reply

Tuning for A4

432 434 436 438 440 442 444 446

View Chords Logout

Octave

0 1 2 3 4 5 6 7 8

Note

A	B ^b	B	C
D ^b	D	E ^b	E
F	G ^b	G	A ^b

Add

Selected Notes in Chord Clear

Generate Graph

Cancel Submit

Clement Guest 5 minutes ago

how about enharmonic spellings?

Reply

Clement Guest 8 minutes ago

maybe less specific than this

Reply

Tuning for A4

432 434 436 438 440 442 444 446

View Chords Logout

Octave

0 1 2 3 4 5 6 7 8

Note

A	B ^b	B	C
D ^b	D	E ^b	E
F	G ^b	G	A ^b

Add

Selected Notes in Chord Clear

Generate Graph

Harmonic Series of Chord 1

Cancel Submit

Clement Guest 28 seconds ago

show notes

Reply

Clement Guest 16 minutes ago

will there be a chord name here?

Reply

Final Prototype:

Before option selection

Tuning for A4 [View Chords](#) [Logout](#)

432 434 436 438 440 442 444 446

Octave

0 1 2 3 4 5 6 7 8

Note

A	B ^b	B	C
D ^b	D	E ^b	E
F	G ^b	G	A ^b

Add

Selected Notes in Chord [Clear](#)

Generate Graph

After option selection

Tuning for A4 [View Chords](#) [Logout](#)

432 434 436 438 440 442 444 446

Octave

0 1 2 3 4 5 6 7 8

Note

A	B ^b	B	C
D ^b	D	E ^b	E
F	G ^b	G	A ^b

Add

Selected Notes in Chord [Clear](#)

B^b D^b

Generate Graph

Result with graph

Tuning for A4 [View Chords](#) [Logout](#)

432 434 436 438 440 442 444 446

Octave

0 1 2 3 4 5 6 7 8

Note

A	B ^b	B	C
D ^b	D	E ^b	E
F	G ^b	G	A ^b

Add

Selected Notes in Chord [Clear](#)

Generate Graph

Harmonic Series of Chord

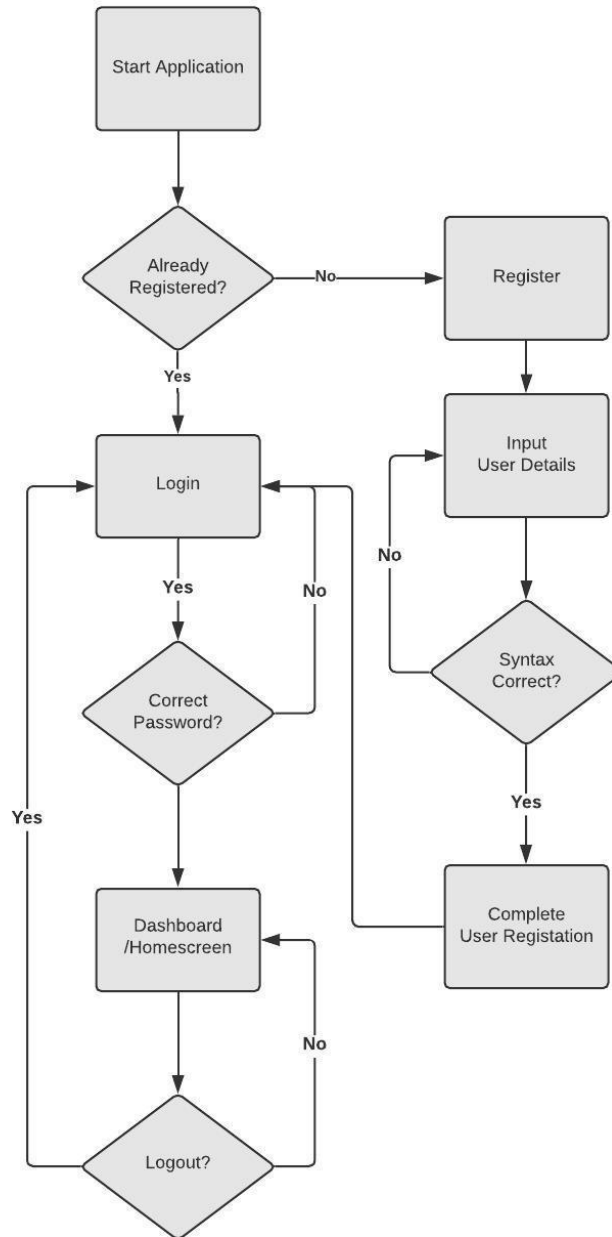
Amplitude

Frequency (Hz)

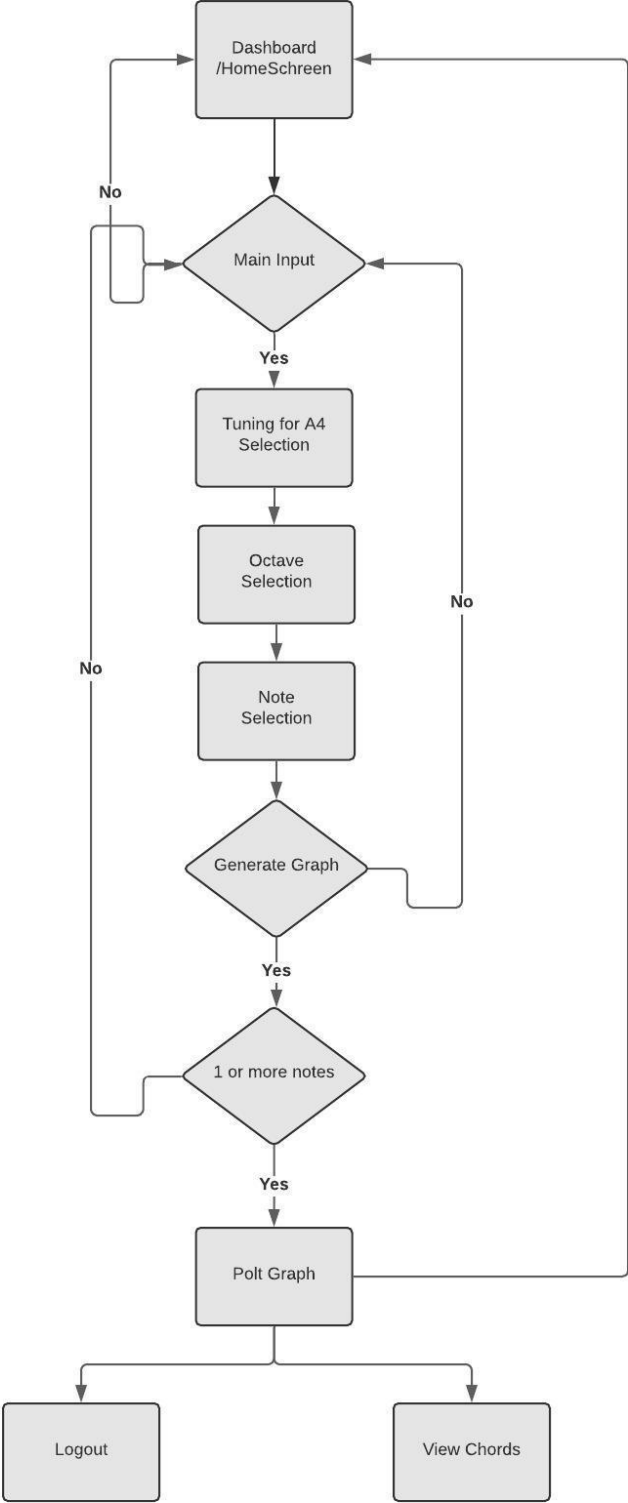
Processing Flowchart

The flowcharts outline the overall logic of the program.

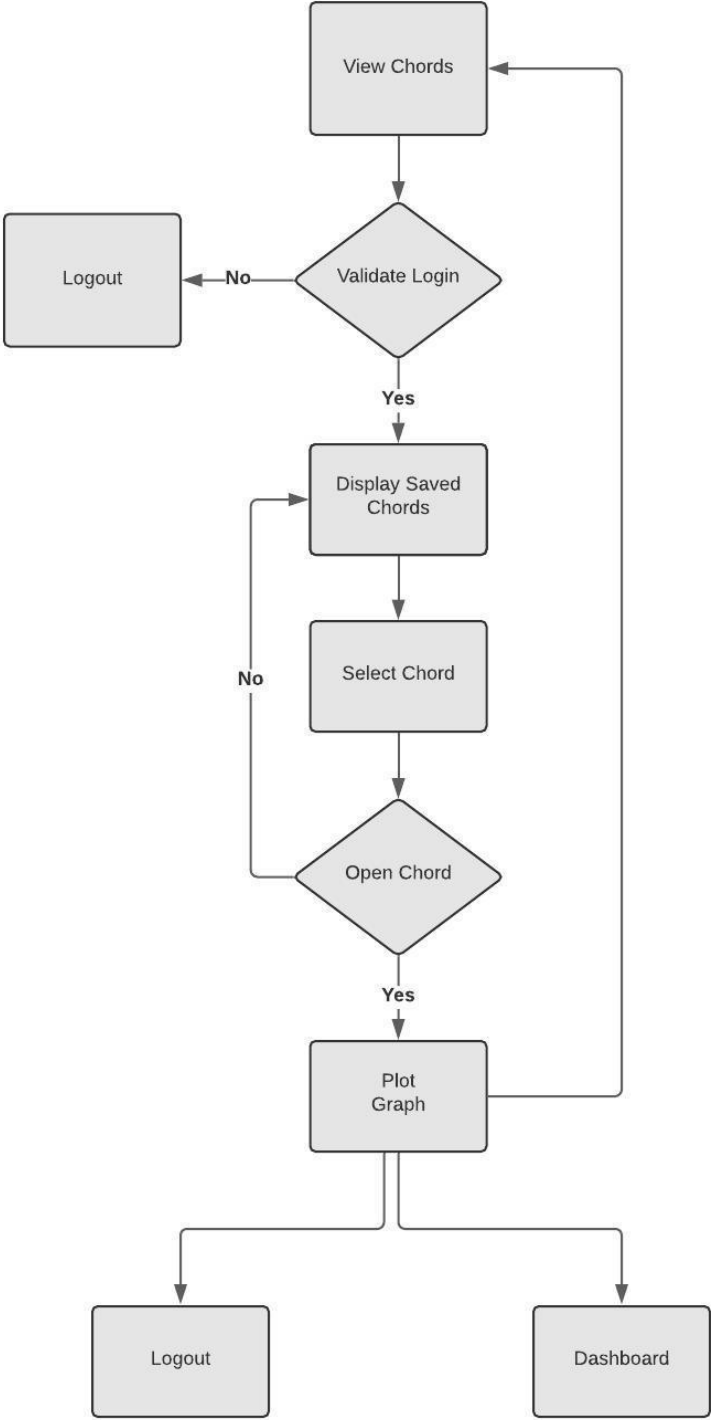
Account ownership diagram



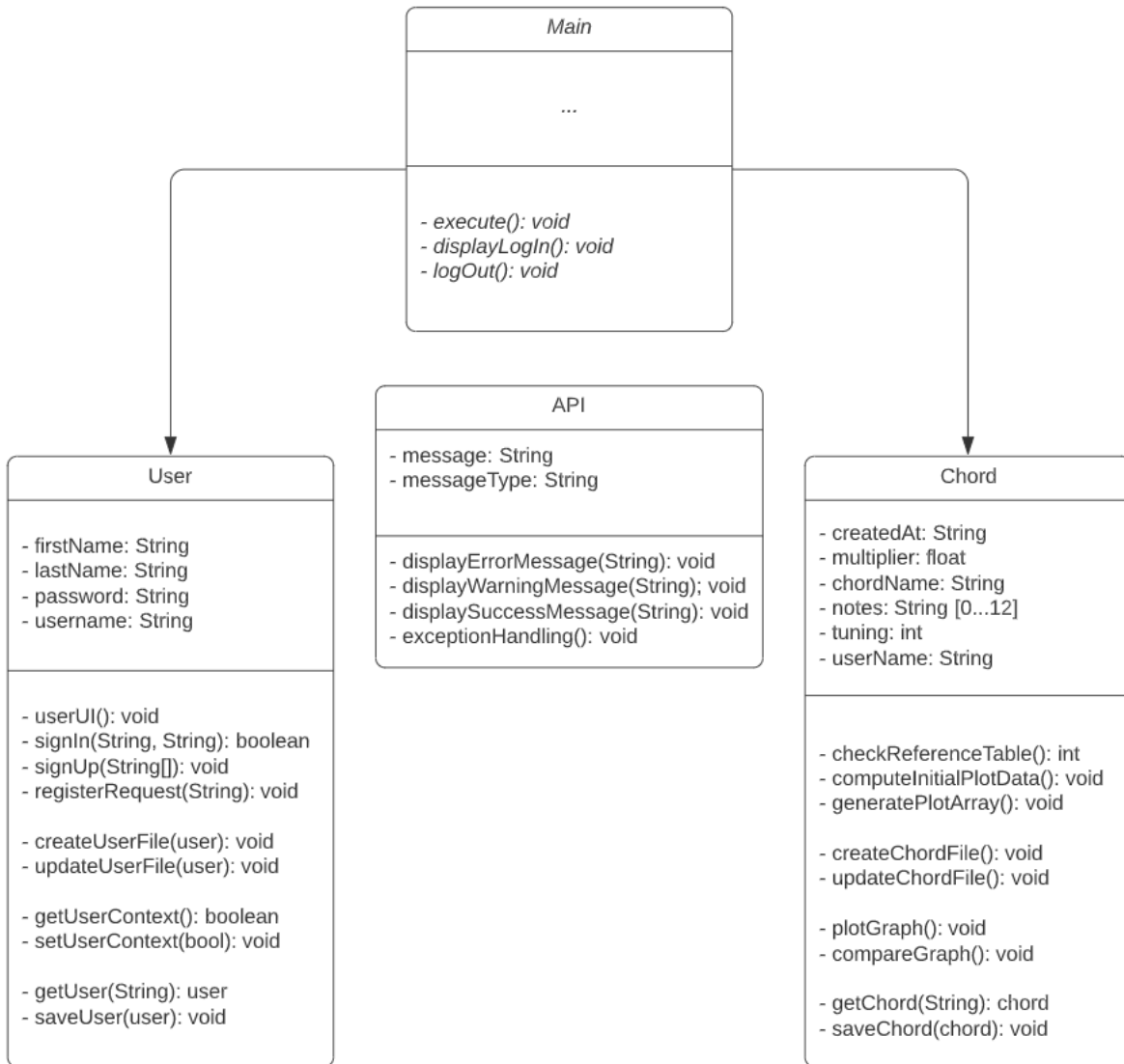
Dashboard Flow diagram



Data Retrieval Flow diagram



Class Diagram



Pseudocode

The following pseudocode methods are written to understand the basic functionality of the main methods that are needed in the development of the program.

Compute Initial Plot Data

```
//computes data for plot based on notes in chord and reference table
void computeInitialPlotData
    var plotData
    accept multiplier, tuning, octave, notes
    validate user inputs

    if valid = true:
        check corresponding reference table
        retrieve reference data
        for each user input:
            calculate plot data using reference data and user input
            //if values have same frequency, stack
            append to plotData

    else if valid = false:
        display error

Show dashboard, compare, logOut
```

Generate Plot Array

```
//preparing to generate graph by creating a plot array
void generatePlotArray
    var plotArray
    accept plotData
    calculate range of plot
    prepare plotArray //generate empty array with range of plot

    for each data in plotData:
        insert into plotArray

//ready to pass plotArray to plotGraph
```

Plot Graph

```
void plotGraph:
    accept plotArray
    prepare graph:
        set x axis = frequency
        set y axis = amplitude
        set legend
        //assign colors to legend according to different notes
        set title
        provide options for zoom, save

    plot graph

    if zoom = true:
        set x axis = frequency
        set y axis = amplitude
        set legend
        //assign colors to legend according to different notes
        set title

    if save = true:
        save graph
```

Retrieve Chord

```
Boolean UserLoginStatus = true //to keep track of current user

display list of saved chords data
accept chord selection from user

if syntax validation = true:
    display notes in array
    plot graph

else if file open validation = false:
    display file open error;

else if generate graph = true:
    compute plotting data and graph

Show Dashboard, Logout
```

Save Chord

```
Boolean UserLoginStatus = true //to keep track of current user

accept chord selection from user

if syntax validation = true:
    format chord data
    save chord data to file

else:
    display file save error;

Show Dashboard, Logout
```

Chronological Development Plan

- Develop UI/UX based on mockup design (1 week)
 - Develop UI/UX components
 - Develop UI/UX classes
 - Implement comments for understandability and extensibility
- Develop code for input functions (2 weeks)
 - Develop for frequencies
 - Acquire reference table - convert to useable data file format
 - Develop data retrieval
 - Develop data formatting
 - Develop input configuration
 - Develop for Octaves
 - Develop for Notes
 - Develop for Array of Notes
 - Store array of notes in chord
 - Display array of notes as selected
 - Clear array of notes
- Develop code for chart (2 weeks)
 - Import plotting library
 - Data retrieval from file
 - Develop graphic design code
 - Configure font, colors, other appearance settings
- Develop file save and retrieve features (2 weeks)
 - Develop Save function
 - Develop Retrieve function
 - Interaction with json CSV filetype

Testing Plan

- Unit Testing
 - Test UI/UX functionality
 - Click buttons - debug GUI code if not functioning
 - Check for successful actions - fix malfunctioning methods
 - Try invalid inputs in texts fields - implement try/catch
 - Null inputs
 - Special/illegal characters
 - Test reference table functionality
 - Correct file format if not being read properly
 - Debug code if file not being read
 - Test chart drawing functionality
 - Use known values to compare against an externally created “correct” graph. Check graph plotting code if inaccuracies are found.
 - Test file storage and retrieval functionality
 - Ensure file generation
 - Check for version of file interaction libraries if an error occurs
 - Ensure data is appended to user and chord files as intended
 - If shown otherwise, debug saving and retrieval code
 - Test math functionality
 - Validate output values. check algorithms if unexpected results occur
 - Negative values
 - Unexpectedly large values
- Integrated Testing
 - Test all the functionality together with known data
 - Compare to known results calculated externally using said known data. Review code if inconsistencies occur
 - Test all the functionality together at datapoint extremes
 - Test functionality at maximum and minimum values. Fix code if errors occur. If maximum and minimum values are not sensible, a sensible result should not be expected; these cases are not errors.
- Functionality Testing
 - Check interaction between GUI and backend. If this is a communication error, review code. Otherwise, check if code files are in correct directories.
 - Registration
 - Login
 - Graph Generation
 - Save and retrieve functionality
 - Chords
 - Users

- User Acceptance Testing
 - Ensure that features requested by the user are provided.
 - Review user feedback and adjust code accordingly.