

Final Presentation



ECOS^S



DENDRO-DAWGZ

| Growing A Brighter Future |

Mentor:

Tayyaba Shaheen

Clients

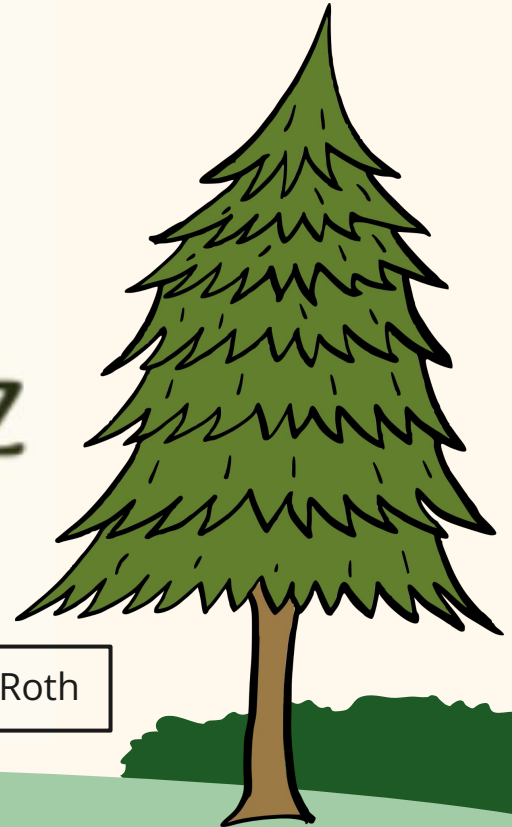
Andrew Richardson

Mariah Carbone

George Koch

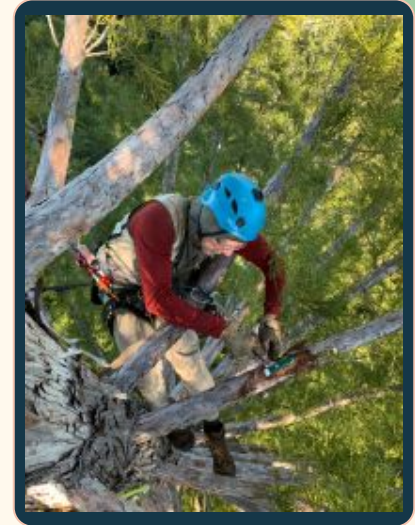
Austin Simonpietri

Zachariah Derrick, Asa Henry, Niklas Kariniemi, Nile Roth



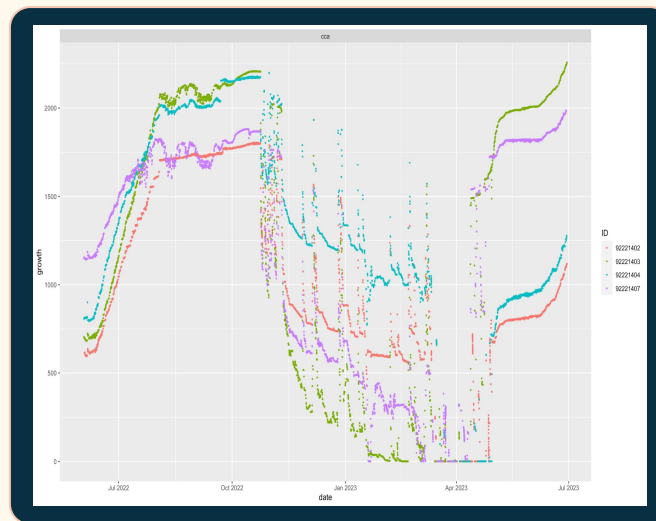
Problem Statement

- Must use laptop while in a tree
- Have to use two hands
- Not the most reliable software
- Can only view one dataset at a time
- Have to share data via google drive



Solution Overview

- Develop an Android application
 - Portable and affordable
 - Able to use with one hand
- Display several dendrometers on the same graph
 - Interactive graphs
 - Comprehensive data
- Export data to the cloud
 - Data and file sharing
 - Viewable from anywhere



Functional Requirements

Data Visualization

Multi-Layered Graphs

Data Reading

Saving Created Graphs

Data Merging

Sharing Created Graphs

Data Storage

Deleting Created Graphs

User Authentication

Non-functional Requirements

Swift Rendering Speeds

Interactive Graphs

Simple & Aesthetic

Secure

Reliable

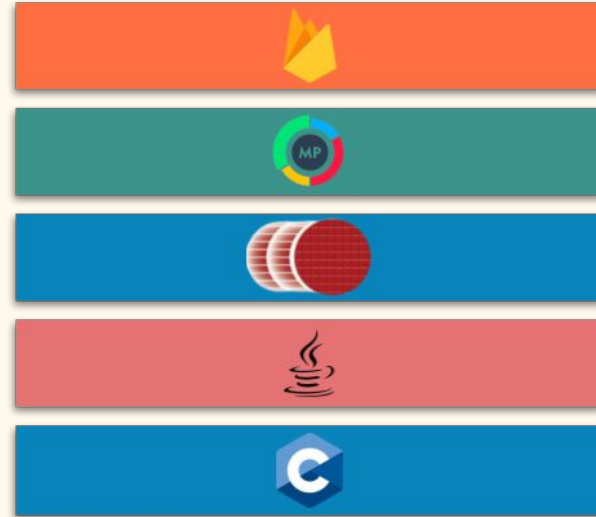
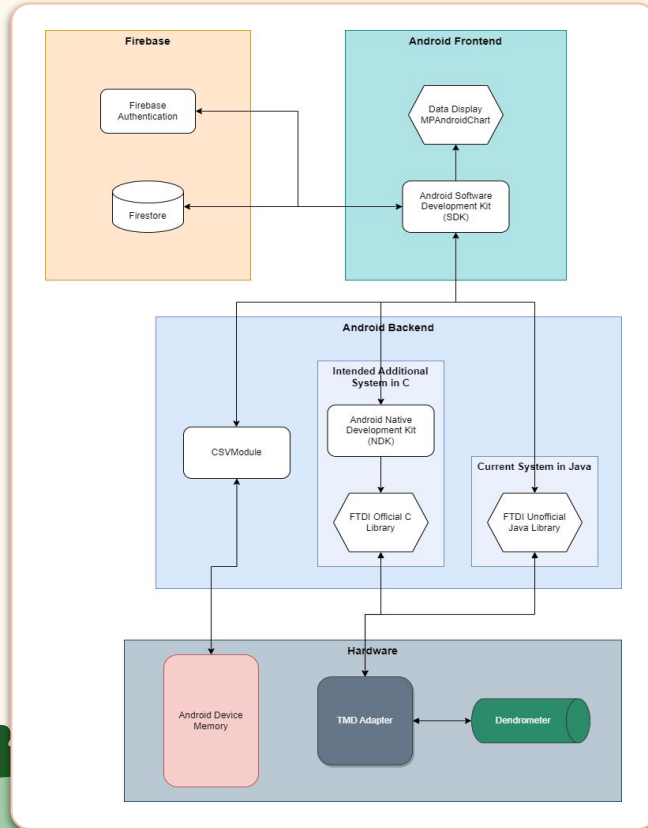
Environmental requirements

- A wired connection is necessitated since the dendrometer does not have a receiver for wireless connection
- The use of an FTDI chip necessitates using the Java and C version of the FTDI library



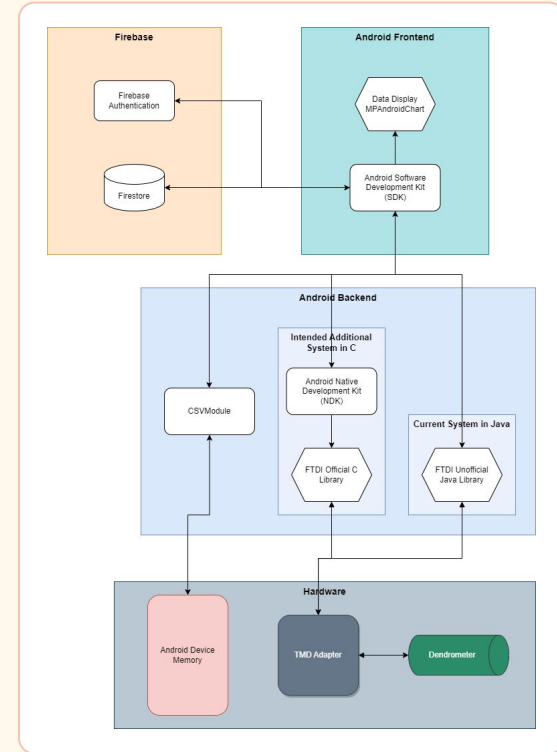
android

Architectural Overview



Implementation Overview

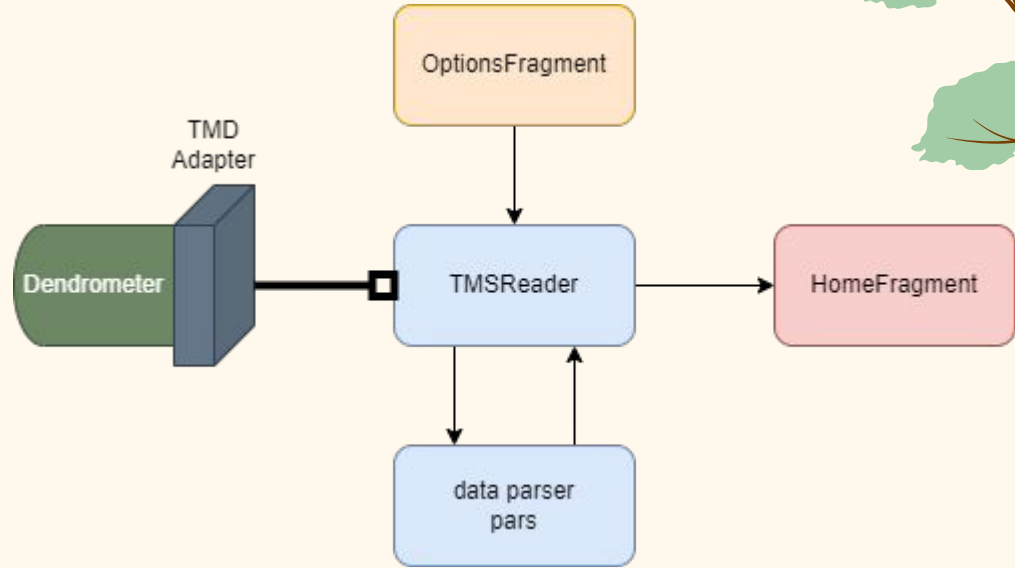
- Core features of the application are implemented
 - Android application developed
 - Hardware connection and complete data retrieval
 - Merging several datasets together
 - Data visualization of both singular datasets and merged datasets
 - Authenticating users
 - Uploading data to a cloud database
 - Sharing data between users using the cloud
- Final features
 - Graph visualization of data analysis
 - Testing the software and monitoring for bugs
 - Implementing user feedback from usability tests



Downloading Data

Dendrometer => TMD Adapter => TMSReader

- The TMD Adapter uses an FTDI chip and transports data to the application
- The TMSReader translates the data to a more readable format



OptionsFragment => TMSReader

- OptionsFragment allows the user to select and input bookmarks

TMSReader => pars

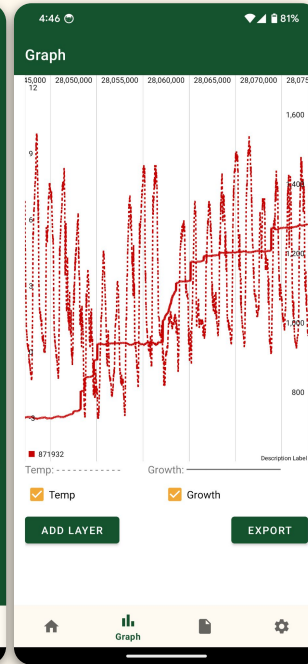
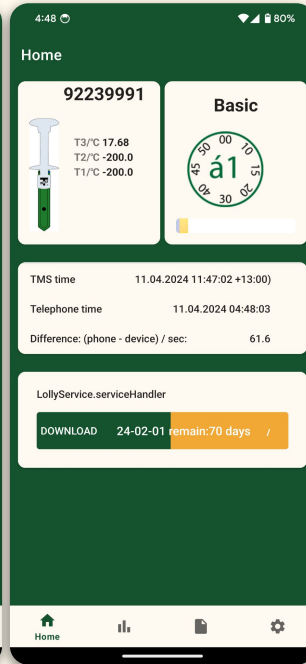
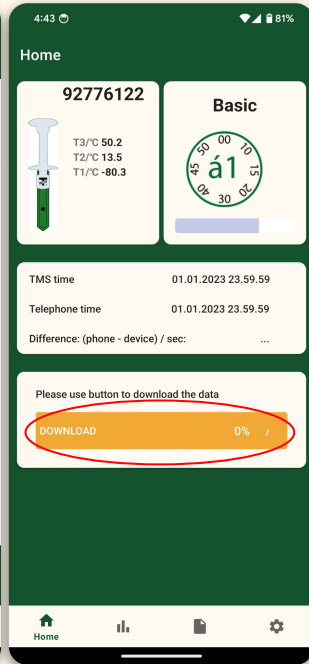
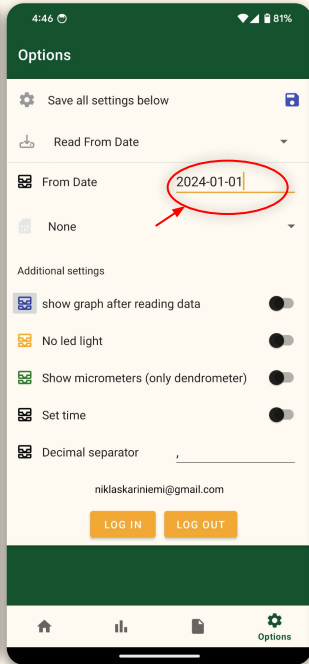
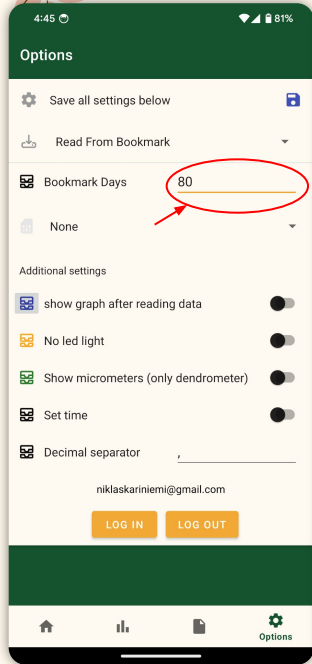
- Pars parses the data into java objects that are more useful

TMSReader => HomeFragment

- The HomeFragment takes the measurement objects and displays the progress and helps write the CSV file

```
<<@ =&93^C0%01.82 TMSx2
>> #@#=#95147321
>> C
<<@C=2024/02/15 16:12:38-12
>> C
<<@C=2024/02/15 16:12:38-12
>> C=2024/02/15 16:12:31-12
<<@C=2024/02/15 16:12:31-12
>> C
<<@C=2024/02/15 16:12:31-12
>> ISPACE
<<@ =&93^C0%01.82 TMSx2
>> W
<<@W.
>> Q Q
>> L
<<@L=00
>> P
<<@P=$005770
>> B
<<@B=$000080
>> S=$000000
<<@S=$000000
>> S
<<@S=$000000
>> D
<<@E=$000000;&,&93%01.82#951473
>> D
<<@DD2024011811054404@E=$0000
```

Downloading Data



	A	B	C
1	1;		
2	871932;1;1;		
3	871932;		
4	1:2023.05.01 00 875;-200;-200;728;206;0;		
5	2:2023.05.01 00 875;-200;-200;728;206;0;		
6	3:2023.05.01 00 75;-200;-200;728;206;0;		
7	4:2023.05.01 00 6875;-200;-200;728;206;0;		
8	5:2023.05.01 01 75;-200;-200;728;206;0;		
9	6:2023.05.01 01 0625;-200;-200;729;206;0;		
10	7:2023.05.01 01 8125;-200;-200;729;206;0;		
11	8:2023.05.01 01 5;-200;-200;729;206;0;		
12	9:2023.05.01 02 125;-200;-200;729;206;0;		
13	10:2023.05.01 0 75;-200;-200;730;206;0;		
14	11:2023.05.01 0 25;-200;-200;729;206;0;		
15	12:2023.05.01 02:45:0;1;-200;-200;730;206;0;		
16	13:2023.05.01 03:00:0;1;-200;-200;731;206;0;		
17	14:2023.05.01 0 875;-200;-200;730;206;0;		
18	15:2023.05.01 0 75;-200;-200;730;206;0;		
19	16:2023.05.01 0 75;-200;-200;730;206;0;		
20	17:2023.05.01 0 625;-200;-200;730;206;0;		
21	18:2023.05.01 0 375;-200;-200;730;206;0;		
22	19:2023.05.01 0 375;-200;-200;731;206;0;		
23	20:2023.05.01 0 3125;-200;-200;730;206;0;		
24	21:2023.05.01 0 25;-200;-200;730;206;0;		
25	22:2023.05.01 0 375;-200;-200;730;206;0;		

Sheet1



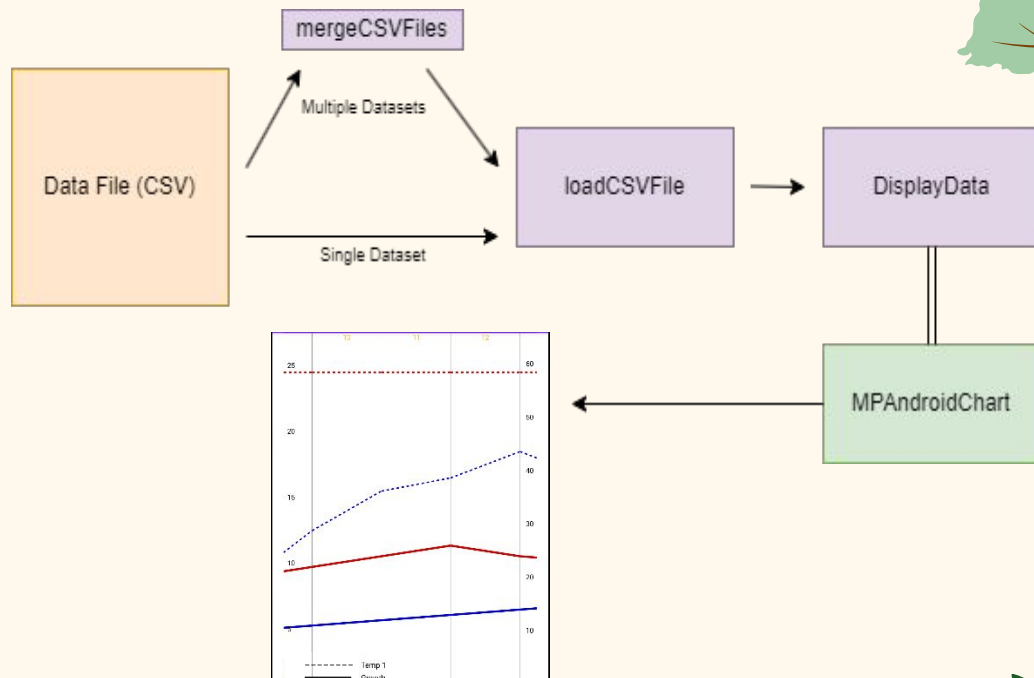
Graph Visualization

LoadCSVFile

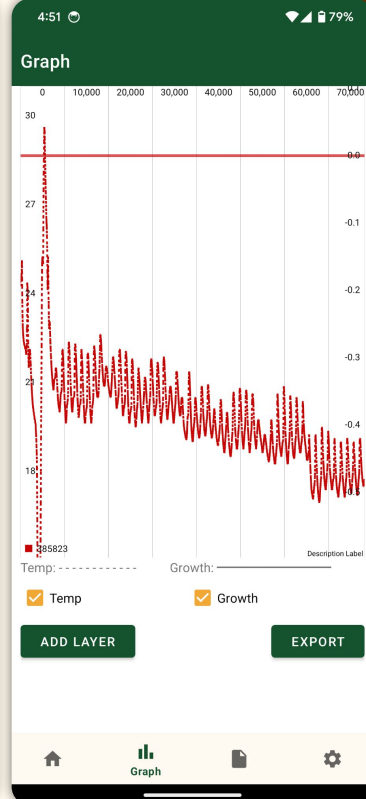
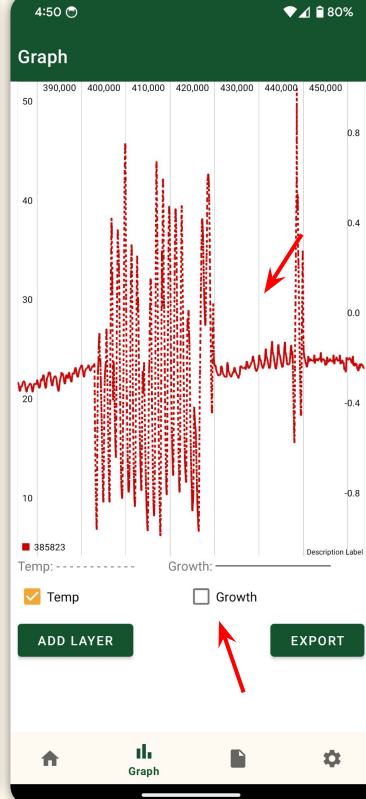
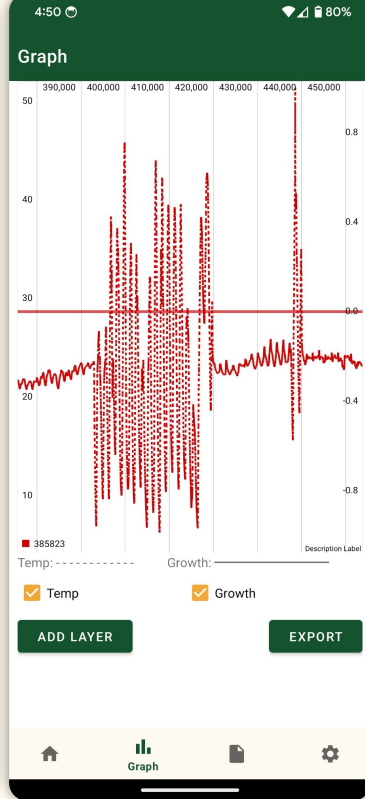
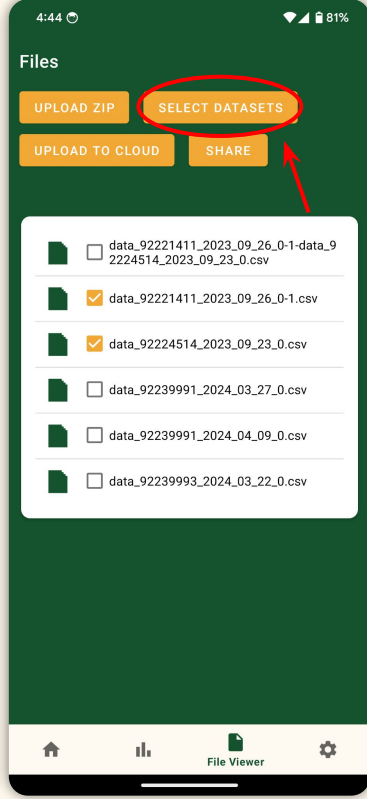
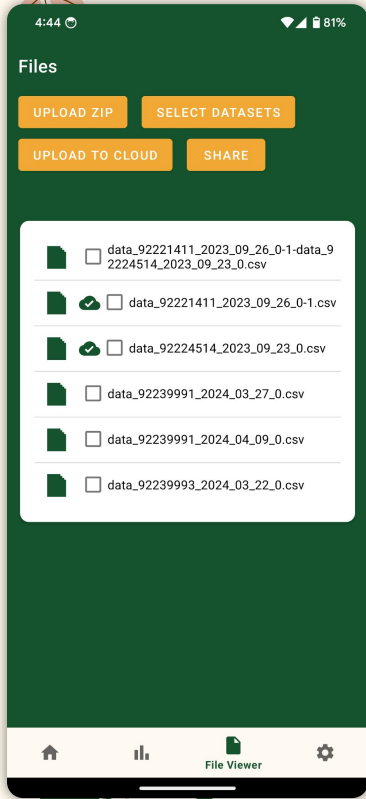
- Converts CSV file into arrays
 - Temperature measurements
 - Compression measurements

DisplayData

- Utilizes MPAndroidChart library
 - Inputs arrays as graph entries
 - Sets color of lines
 - Outputs line onto view



Visualizing Data



Merging CSV Files

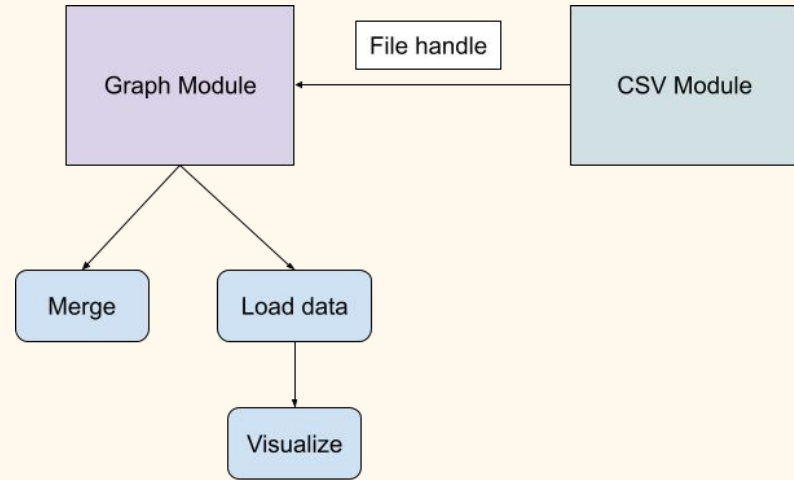
- CSVReader -> CSVFile
 - CSVFile provides simplified API to interact with files
- GraphFragment
 - Merges N data sets into a single data set
 - Provides header for metadata

Example of header single data set file:

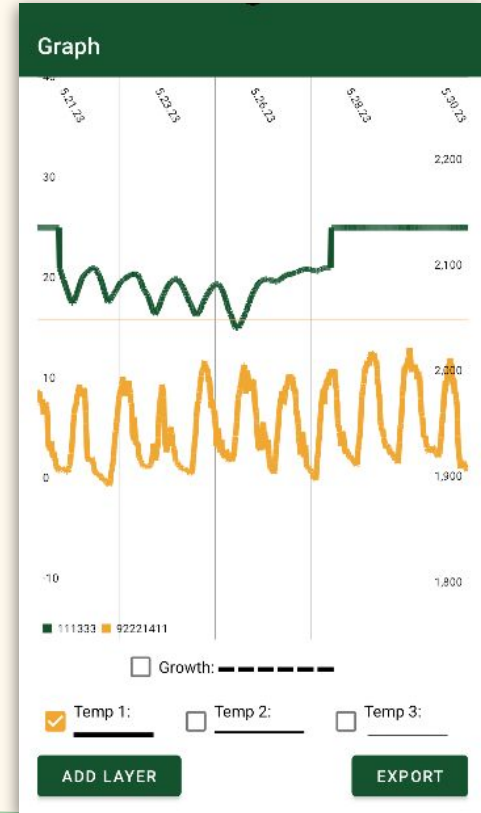
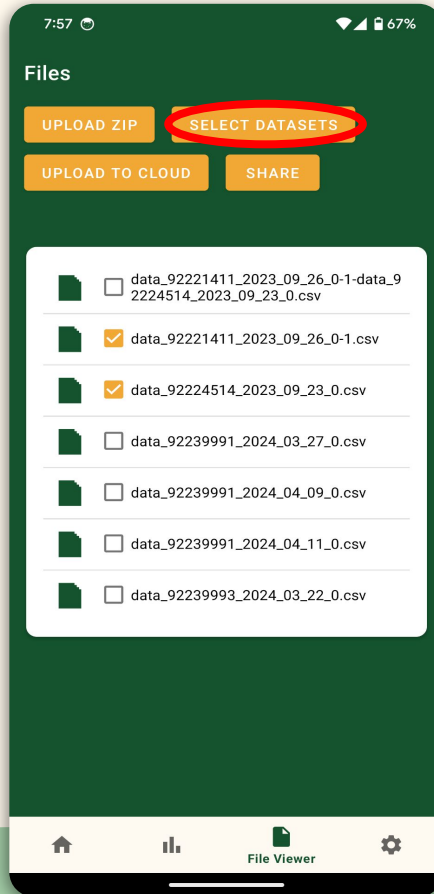
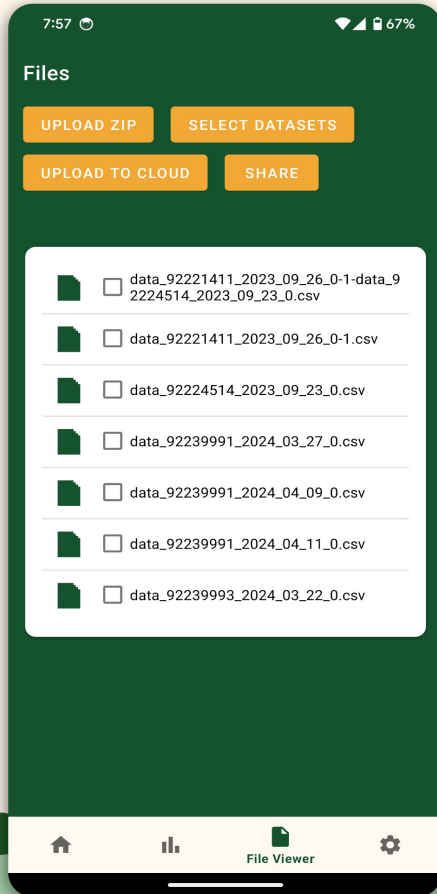
```
1;  
83974;1;1;  
83974;  
// data
```

Example of header merged data set file:

```
3;  
83974;1;1;  
72941;1;0;  
13621;2;1;  
// first data set
```

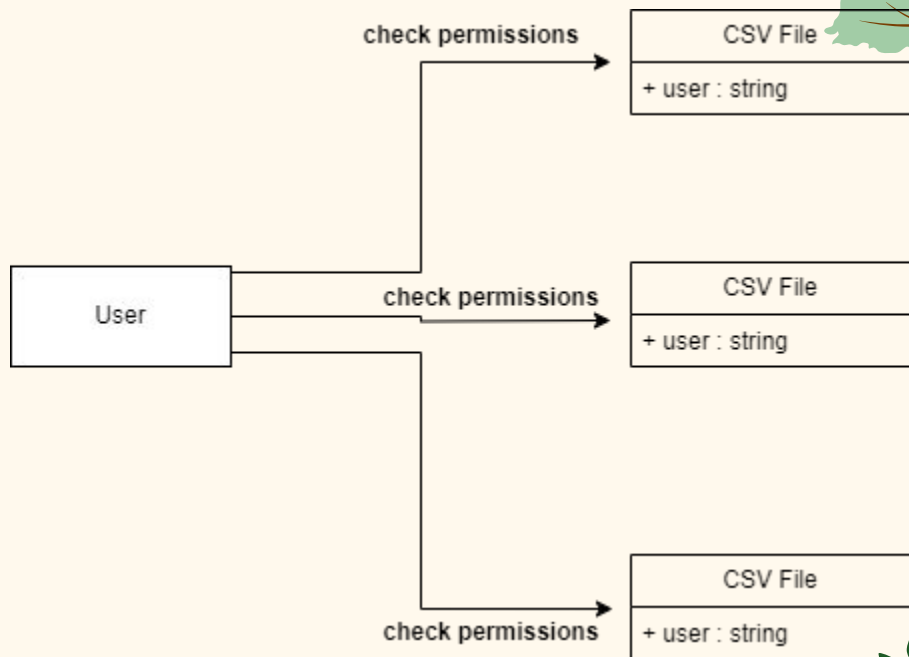


Merging CSV Files

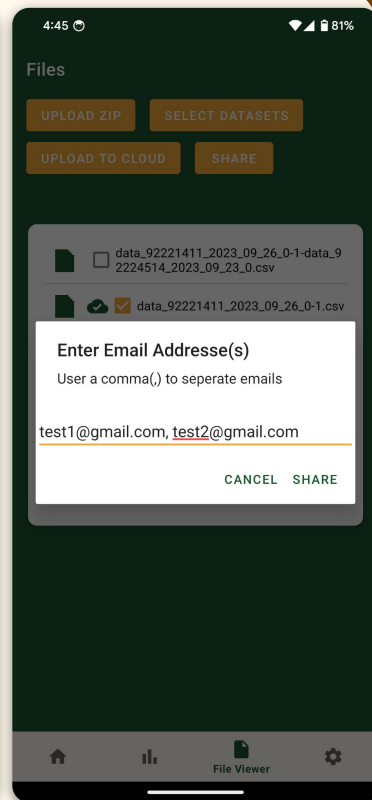
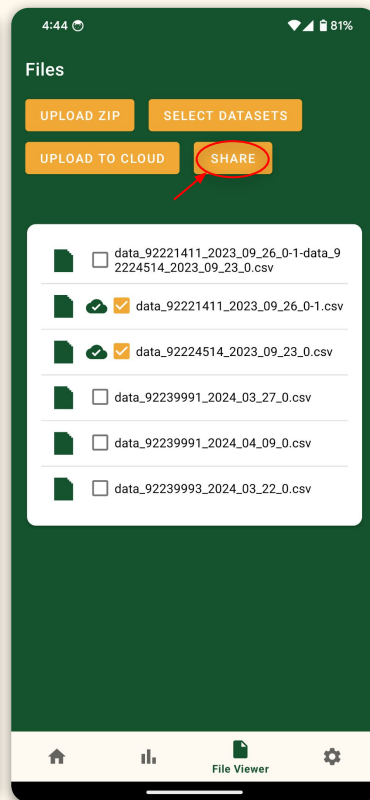
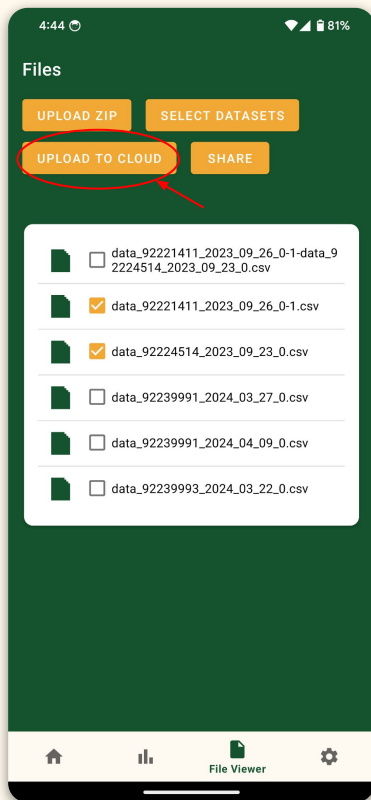
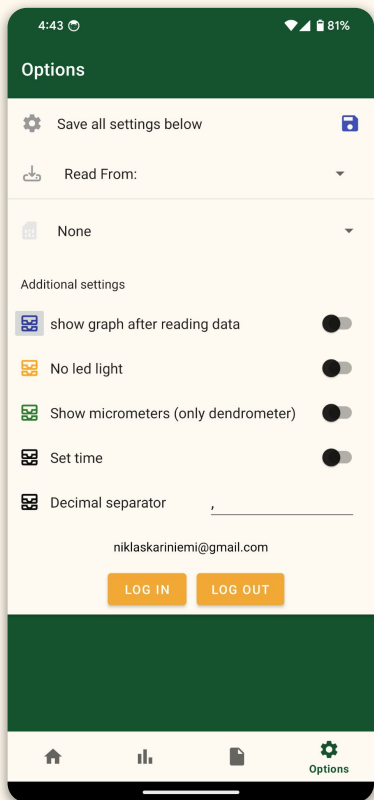
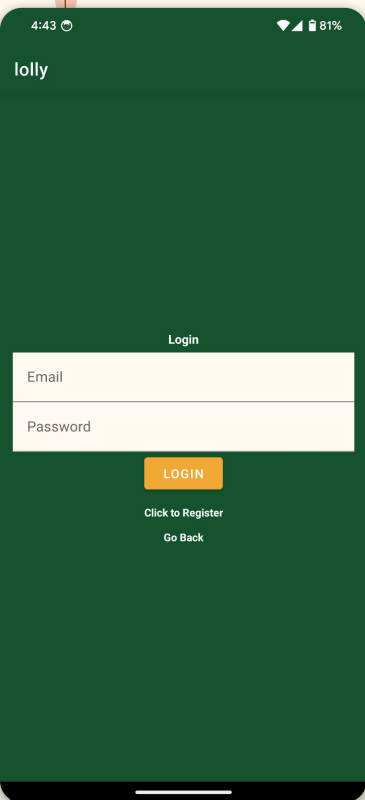


Cloud Export

- User Authentication is implemented via email and password
- Storing on Firebase with email, password, and a unique id
- To see data stored on Cloud Storage, must be signed in
- Go through each file
- Check if current user is in the allow list
- Allow list is in files metadata
- Anyone with access to the file can share



Cloud Export



Challenges and Resolutions



	Merging	Graph	Cloud	Backend
Challenge:	Support two file formats our clients use; support files from Lolly software (for Windows)	Hard to differentiate 4 lines for each dataset in graph	Uploading a csv file to a database would require uploading every single row and column.	Acquiring all device commands, and accessing device functions using them
Resolution:	Provide dedicated functions for converting between formats; use additional methods to find serial number	Dataset identified by color Growth line is dashed Temperature lines are differentiated by line weight	Use cloud storage instead of a database. Allows for uploading csv files.	Got proper command list from TOMST engineer and got advice on implementation



Schedule



We are here

Tech Feasibility Process



Requirements Specifications



IMPLEMENTATION

Downloading Data



Merging Datasets



Graphing UI



Exporting/Storing



Testing



Final Documentation



Testing

Unit testing

- pars.java
 - Used to parse incoming data
- CSVFile.java
 - Used to interact with filesystem and change files
- GraphFragment.java
 - Used to visualize data from dendrometers
- ListFragment.java
 - Used to populate list in File Viewer page

Integration testing

- Visualizing and merging data
 - Visualizing small and large datasets, and merging them
- Login Flow
 - Passing user data across different views

Usability testing

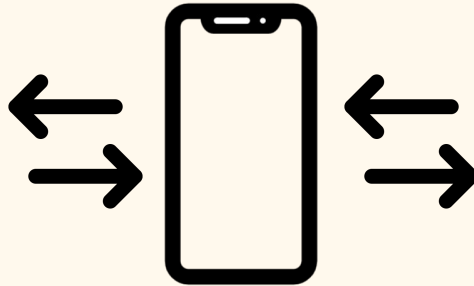
- Data reading
 - Data download is straightforward
 - Bookmarking and reading from date
 - Info presented to user is useful
- Visualizing data
 - Able to navigate to File Viewer page
 - Easily able to select/deselect files
 - Can navigate straight to the button
- Merging data
 - Merging is intuitive
 - Converting file formats is easy
 - Finding the file select page is easy
- Exporting data to the cloud
 - Able to easily navigate to File Viewer page
 - Intuitive on how to select files
 - Easily understood on how to upload files

Future Work

Release Date: May 5, 2024



Point Dendrometer
Tree Growth



TMS-4
Soil moisture



Conclusion




Our clients want a **better solution** to the current way of collecting data from dendrometers. Having to carry a laptop into a tree makes process harder and more dangerous

Our clients use tree growth data to study the relationship between trees, other environmental players, and our changing climate; this application facilitates the research our clients and others carry out; our application is integral to understanding and **beating climate change**

We will create a mobile application for Android which will have the ability to **read** in data, **store** data, **merge** data, **visualize** data, as well as **share** data with others using a cloud solution

Our application will *impact* researchers and professionals alike by making conditions **safer** for collecting data, as well as making the process go **faster** than previously

Clients are very happy with the work we have completed and it has been a great experience for us!



Thank You!

