

Requirement:

Findings:

1) Researching and installing Flutter

<https://www.youtube.com/watch?v=tun0HUUHaDuE>

<https://docs.flutter.dev/get-started/install/windows>

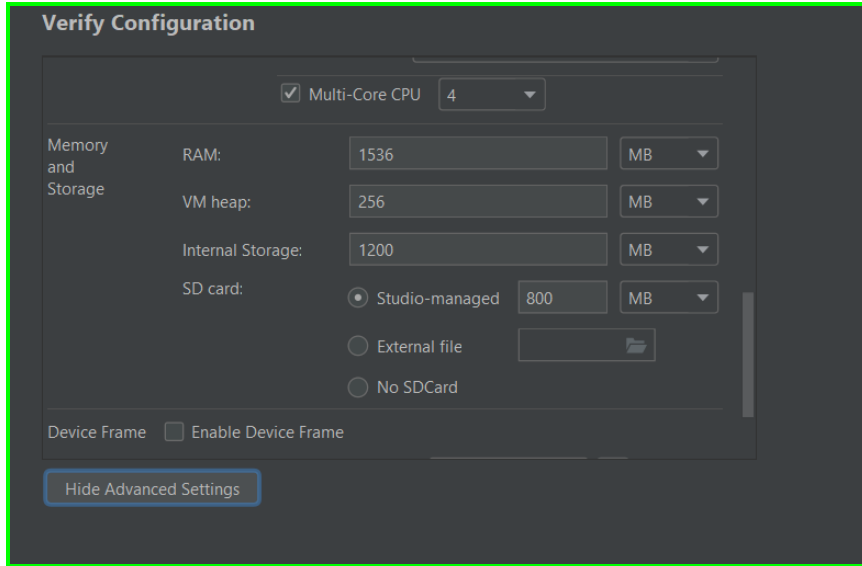
Choosing a language was a decision that needed time and discussion. We originally wanted to develop for IOS devices, however we weren't sure if this was something we wanted to commit to. We knew swift has traditionally been used for IOS development and Java is used for android development. We wanted to develop an app that would be cross compatible for both. This narrowed it down to React native and Flutter. The decision to go with flutter was just out of chance. Both of the options achieve the goal of running programs on both IOS and android which is what was important to us.

At first, installing flutter proved to be a challenge as I planned on using WSL2 to run the program. However, since this is a hybrid program of linux and windows. The installation was not as straightforward as running purely on windows or purely on linux. After a series of failure installations, I decided to commit to a windows focus installation. Initially we expected to use html and css for the front end. However, we quickly realized that Dart could control all the files needed for the User Interface. This meant a more meaningful approach as there were less files to keep track of and one less dependency to worry about when it came to connecting all these different platforms together.

To install Flutter, we referred to a Youtube video. The link to the video is as above. To start the process, we had to first go to the Flutter Website and ensure that our devices were compatible to have Flutter downloaded. We also needed to install Windows Powershell, Git for Windows and Android Studio.

We needed to install Android studio to simulate the environment of a phone. This was a fairly straightforward process as all the documentation was provided when we were installing Flutter.

Sprint 2- Findings



The image above is from the SDK manager as a part of the installation for Android Studio

<https://levelup.gitconnected.com/login-page-ui-in-flutter-65210e7a6c90>

<https://blog.logrocket.com/implementing-secure-password-reset-flutter-firebase/>

Connecting MongoDB to Visual Studio and also learning more about using MongoDB:

<https://code.visualstudio.com/docs/azure/mongodb>

<https://www.geeksforgeeks.org/how-to-install-mongodb-for-vscode/amp/>

<https://code.visualstudio.com/docs/azure/mongodb>

<https://www.geeksforgeeks.org/how-to-install-mongodb-for-vscode/amp/>

https://www.tutorialspoint.com/mongodb/mongodb_create_collection.html

These four links are useful to learn more about how MongoDB works. Additionally these links help with connecting MongoDB to visual studio. This is something that our team has been discussing throughout the last sprint.

Sprint 2- Findings

Connecting MongoDB with Flutter:

<https://flutteragency.com/connect-flutter-with-mongodb/#:~:text=Let's%20start%20learning.-.How%20to%20Connect%20Flutter%20With%20MongoDB%20%3F%3F,%3A%20io%2C%20dart%3A%20HTML.>

<https://levelup.gitconnected.com/mongodb-with-flutter-407de79f84e4>

<https://360techexplorer.com/connect-flutter-to-mongodb/>

Connecting Flutter and MongoDB:

<https://360techexplorer.com/connect-flutter-to-mongodb/>

<https://flutteragency.com/connect-flutter-with-mongodb/>

<https://www.youtube.com/watch?v=juKDXPk7kU4>

Connecting Dart and MongoDB:

<https://www.youtube.com/watch?v=Y5X5rdzFScs&t=277s>

MONGODB RELATED FINDINGS

As a group we decided to work with Flutter as our base language and work with mongodb. None of us have ever worked with mongodb before and hence we took this sprint to work on developing our skills. We each selected a topic that interested us and decided to understand and practice the topic. I picked MongoDB as my topic. My first step was to understand how to connect a flutter program to mongodb. On looking through a few slides I saw this. We use `mongo_dart` as the server side library. To import that we use *import 'package:mongo_dart/mongo_dart.dart' show Db, DbCollection;*

Working with mongodb vs SQL.

MongoDB is a NoSQL server. Where data is stored in B0SON documents, and each document is essentially built on a key-value pair structure. MongoDB easily stores schemaless data, making it appropriate for capturing data whose structure is not known. This document-oriented approach is designed to offer a richer experience with modern programming techniques. In our project we would be required to store data such as journal entries. This is a lot of data, and the data is not structured. Hence, I believe MongoDB would be a better fit for our project. The MongoDB databases are also scalable and flexible.

Sprint 2- Findings

How to write queries in MongoDB?

To find a particular entity in the table we use the command find.

Db.peaks.find({"<value_to_be_found>": "<value_to_be_equal_to>"})

Multiple conditions can also be used in a similar syntax followed by commas. To look for more values or similar values we can use an array-like setup.

No joins would be needed or are possible in MongoDB and accessing and extracting data is easy.

Like in SQL we can also filter the number of outputs like the best 3 or the last 3. This can be done using the limit function.

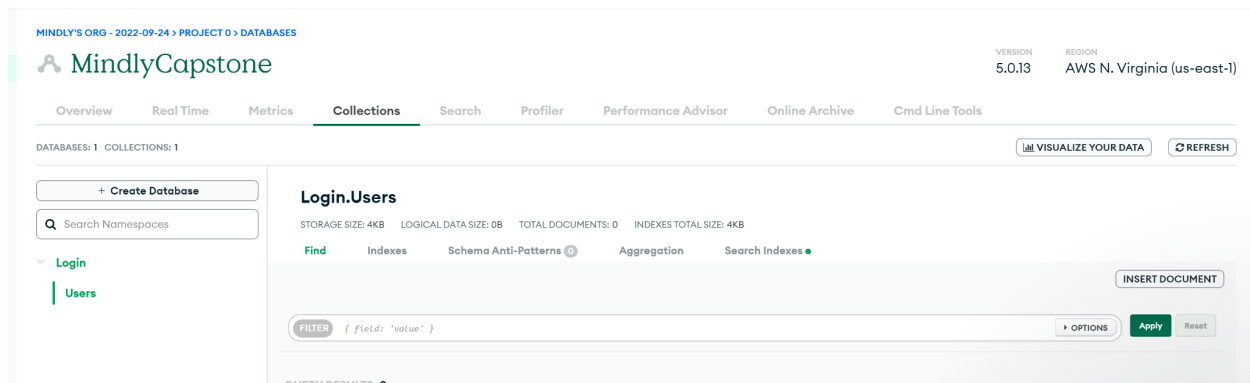
This is a start to understanding and learning MongoDB. It is going to be a learning process as we work through this project but having a fair idea of how it works and understanding a few basic syntax is extremely helpful.

Creating a table in MongoDB

–<https://www.youtube.com/watch?v=l2E2kSM9aIg>

https://www.tutorialspoint.com/mongodb/mongodb_create_collection.htm

After extensive research, we were able to create a login table in MongoDB.



As seen in the image above, we were successfully able to create a table for the user login page. Upon connecting this table with Dart, we will be able to add in columns that will have the username and the password stored for each user. This way, we will be able to keep track of an existing user. If the user logging in is not an existing user, we can prompt them to sign up instead by taking them to a signup page which was implemented in Sprint 1. However, this connection is a goal for future sprints. As we had mentioned in the requirements artifacts our main goal was to show an understanding about what we had learned from the research by creating the users table.