Software Project

Team Name: TeamBot
Project Name: Cave Examination Bot
Group Number: 9

Student Names and IDs:
Adrian Portal Calcines - n01489363
Alfred Dowuona - N01490404
Ali Mohebi - N01477361
Hassan Noorani - N01485518

Table of Contents:

Signatures and Effort Table:

Name	Id	Signature	Effort
Adrian Portal C	n01489363	A.P.	100%
Alfred Dowuona	n01490404	A.D.	100%
Ali Mohebi	n01477361	A.M.	100%
Hassan Noorani	n01485518	H.N.	100%

Project Scope and Goals:

The project scope will be an Android studio application that can display sensor readings gathered from a robot stored in a database. It will have various screens such as main dashboard, configuration screen, data readings and data analysis screen and other minor functionality screens like notifications. We plan to make the app as user-friendly as possible so the user can interact with the data and manage the robot easily. Our goal for this course is to make a functioning app that integrates and modifies a database, displays and manipulates data within the app. Goals like implementing functionalities to a configuration screen, making a dashboard that looks and feels right, having menu items with different features, and overall making an app that can be verified and deployed into the google play store. We can take it further and set up the app so it reads from the sensor data collected from our raspberry pi sensors directly. We will learn and implement security within the app since we are handling sensitive data. Also since we are using the agile method we will adapt and plan scalability to leave room for further development in future courses.

GitHub Repo and Strategy:

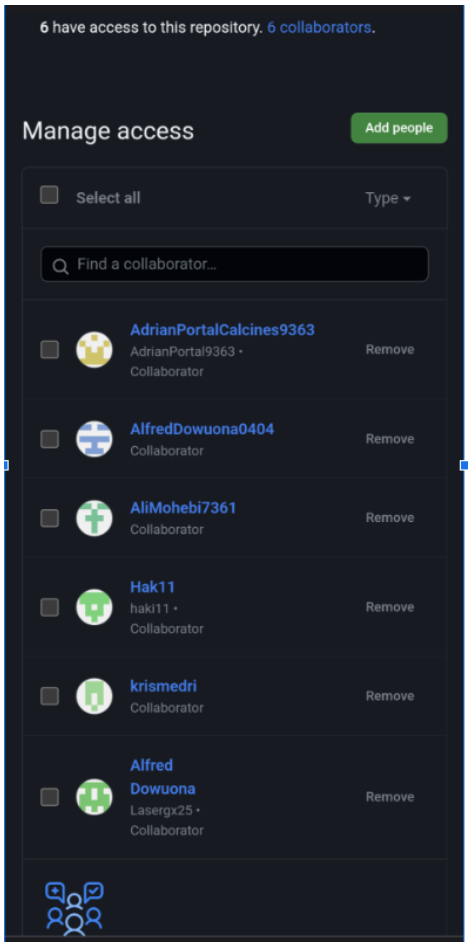
Link:

https://github.com/HassanNoorani5518/CaveExaminationBot

GitHub Strategy:

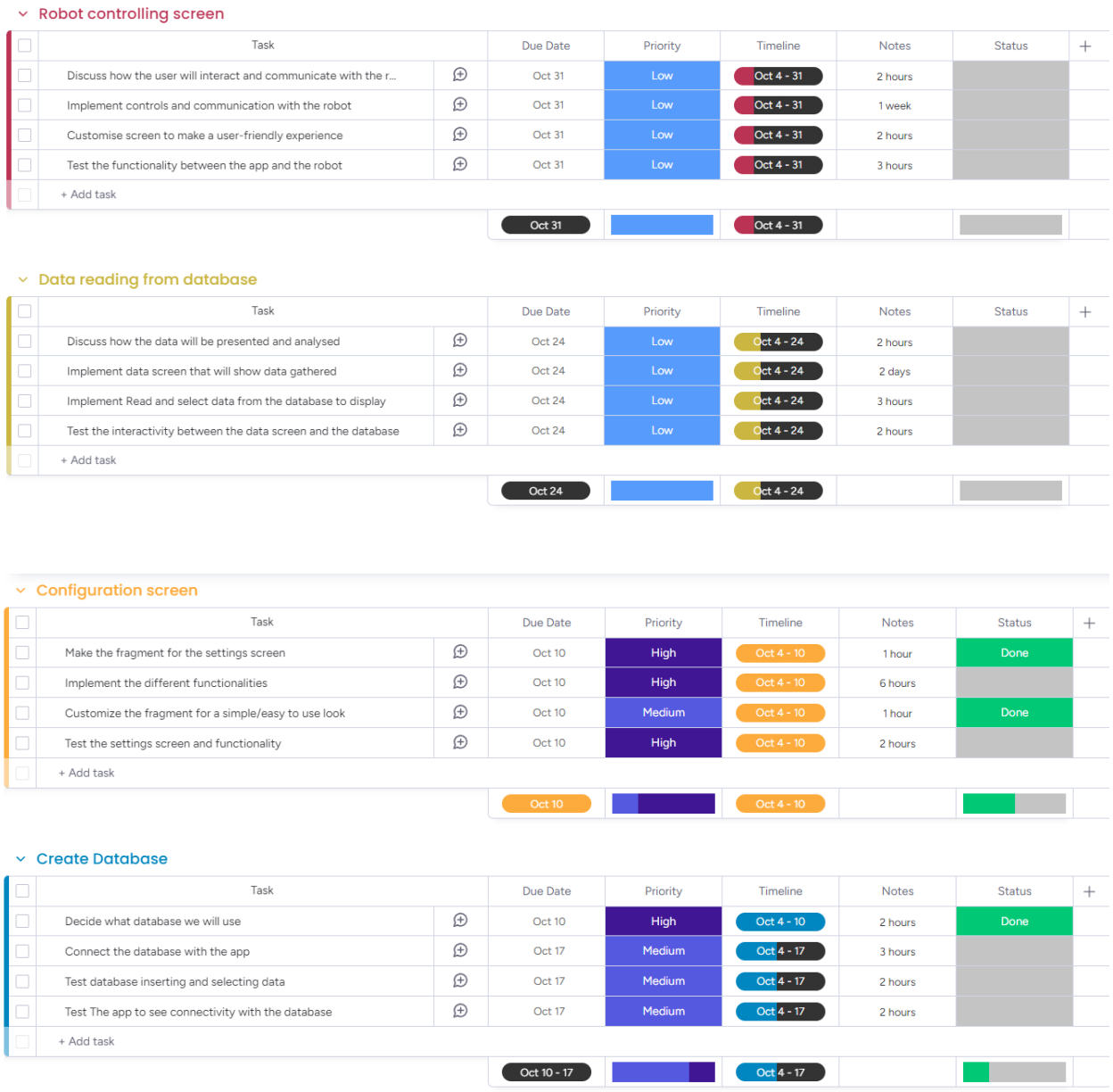
Our approach to the contribution of the project on github starts with setting a time in the day when each member will work on the project. Each member books a time in the week where they alone will work and clone the project. In this period all other members can answer any questions the member working on the software has about any previous code or contributions. Once a member implements their code they will commit the code with an accurate description of the changes made. We work on the deliverable chronically and don't skip a step. Once all tasks for the day and week have been completed each member will clone the repository and perform tests and quality control, exploiting the app to find possible bugs and fixing them.

Screenshot:



Stories and Tasks:

Screenshot:



Robot controlling screen

Task	Due Date	Priority	Timeline	Notes	Status
Discuss how the user will interact and communicate with the r...	Oct 31	Low	Oct 4 - 31	2 hours	
Implement controls and communication with the robot	Oct 31	Low	Oct 4 - 31	1 week	
Customise screen to make a user-friendly experience	Oct 31	Low	Oct 4 - 31	2 hours	
Test the functionality between the app and the robot	Oct 31	Low	Oct 4 - 31	3 hours	
+ Add task					
	Oct 31		Oct 4 - 31		

Data reading from database

Task	Due Date	Priority	Timeline	Notes	Status
Discuss how the data will be presented and analysed	Oct 24	Low	Oct 4 - 24	2 hours	
Implement data screen that will show data gathered	Oct 24	Low	Oct 4 - 24	2 days	
Implement Read and select data from the database to display	Oct 24	Low	Oct 4 - 24	3 hours	
Test the interactivity between the data screen and the database	Oct 24	Low	Oct 4 - 24	2 hours	
+ Add task					
	Oct 24		Oct 4 - 24		

Configuration screen

Task	Due Date	Priority	Timeline	Notes	Status
Make the fragment for the settings screen	Oct 10	High	Oct 4 - 10	1 hour	Done
Implement the different functionalities	Oct 10	High	Oct 4 - 10	6 hours	
Customize the fragment for a simple/easy to use look	Oct 10	Medium	Oct 4 - 10	1 hour	Done
Test the settings screen and functionality	Oct 10	High	Oct 4 - 10	2 hours	
+ Add task					
	Oct 10		Oct 4 - 10		

Create Database

Task	Due Date	Priority	Timeline	Notes	Status
Decide what database we will use	Oct 10	High	Oct 4 - 10	2 hours	Done
Connect the database with the app	Oct 17	Medium	Oct 4 - 17	3 hours	
Test database inserting and selecting data	Oct 17	Medium	Oct 4 - 17	2 hours	
Test The app to see connectivity with the database	Oct 17	Medium	Oct 4 - 17	2 hours	
+ Add task					
	Oct 10 - 17		Oct 4 - 17		

App menu

Task	Due Date	Priority	Timeline	Notes	Status
Discuss what menu items are needed and will be implemented	Oct 10	High	Oct 4 - 10	1 hour	Done
Implement the menu with all items and buttons functionality	Oct 10	High	Oct 4 - 10	2 hours	Done
Add design to the menu, images and placement	Oct 10	High	Oct 4 - 10	1 hours	Done
Test the menu and their functionality	Oct 10	Medium	Oct 4 - 10	3 hours	Done
+ Add task					
	Oct 10		Oct 4 - 10		

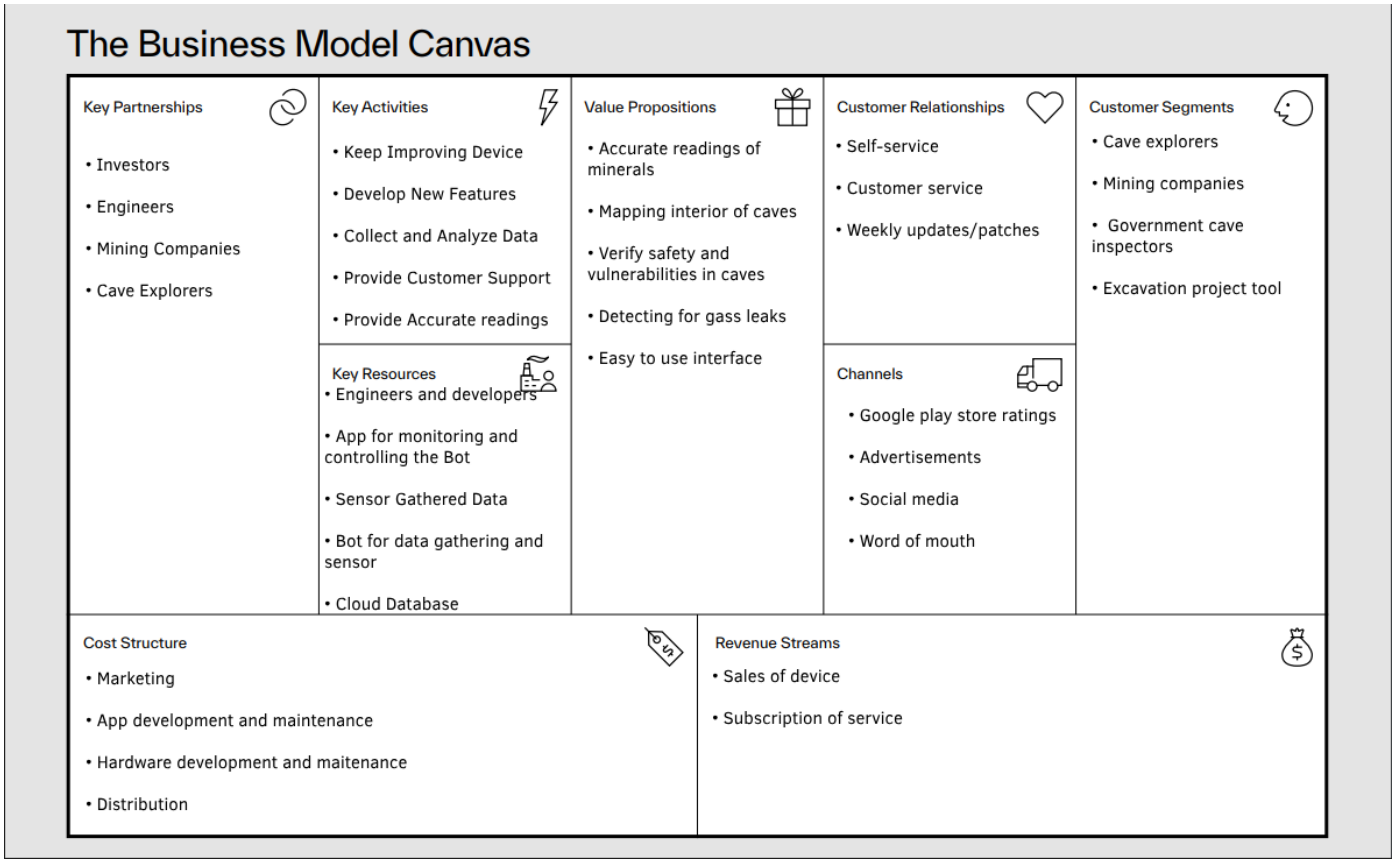
Done Tasks:

For the done tasks, we labeled them completed once they met the DoD criteria. We performed our list of the DoD items on the finished code. We determined it was completed when the code was finished, acceptance of the implementation and functionality was verified by all group members and was green lighted, testing and bug fixing was completed, and lastly the code was efficient and easy to maintain/add on to for future maintenance.

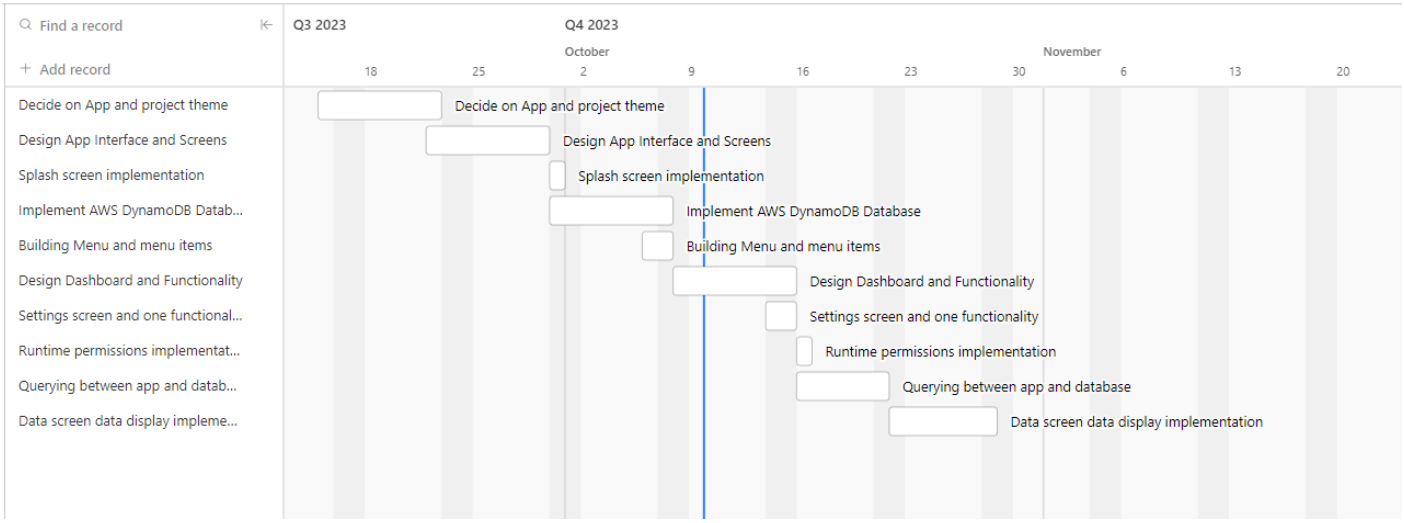
List of DoD items included:

- Acceptance criteria is verified during testing
- Coding tasks completed
- Exploratory testing completed and signed
- Code reviews conducted

Business Model Canvas:

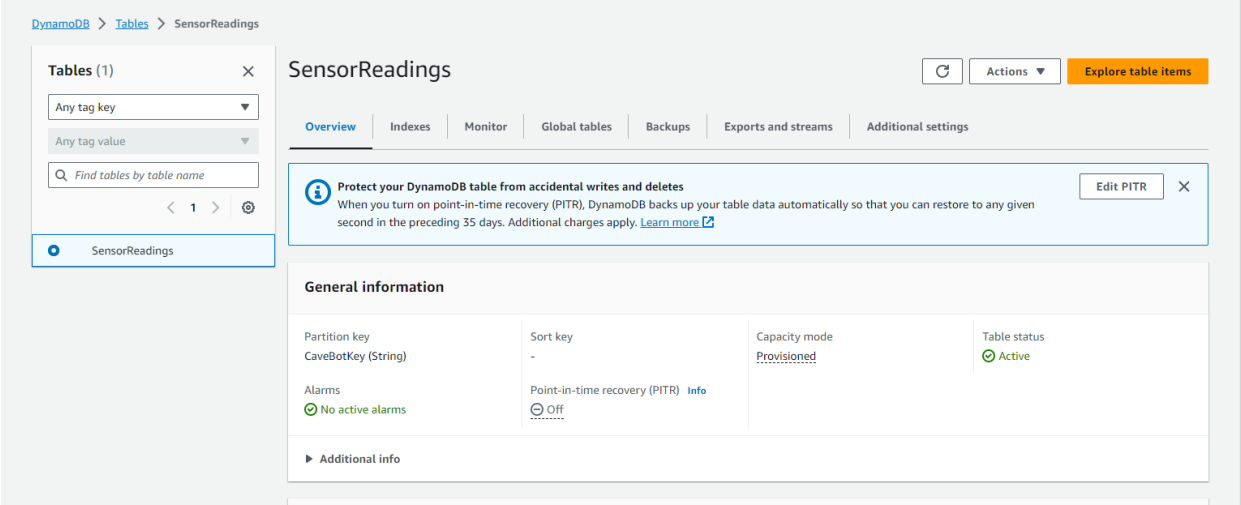


Gantt Chart:



Database Screenshot:

We decided to use the free tier AWS DynamoDB database.



Database plan:

For the database we are using AWS DynamoDB, we created a table named SensorReadings with various fields subject to change depending on requirements. Along with the table we created a partition key that will allow us to use the app to edit and communicate with our database. We will integrate the AWS SDK on our android studio app to allow support and interaction with the AWS service. We set up the AWS SDK with root credentials to get permissions and access to our database. Later on we will create an instance of the DynamoDB client and test inserting and selecting data. Data inserted into the table can be viewed on the AWS web page for clarity and further editing. This database is free and allows us to use enough features to implement our app functionalities.

Additional Features Added:

- Designed the configuration screen, and implemented the lock screen to portrait functionality (no other functionality yet).
- Added 6 more images across all screens with 3 different resolutions.
- Added support for both landscape and portrait layout in the splash and main screen.
- Added the menu feature with 4 options the user can interact with.
- Implemented feature that requires the user's permission to know the location of the device, added screen if permission not granted, and a snackbar displaying whether it was granted or not.
- Added image to the tabs, menu items, and back key dialog.

Meetings Record:

Date	Info
04-10-2023	- Meeting to discuss what database to use - How we were going to implement said database - Overall our thoughts and ideas on the project so far
07-10-2023	- Meeting to discuss the dashboard look/functionality - What the configuration screen was going to have - Expectations on when the code should be done
10-10-2023	- Meeting to talk about the last features needed to be done - Work on the in class presentation and powerpoint - Testing and quality control of the app