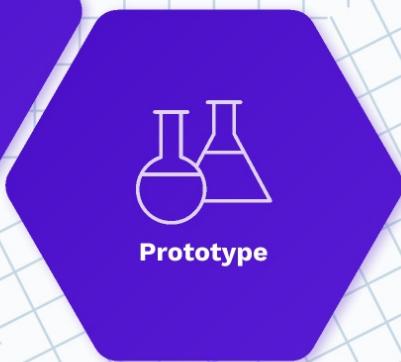
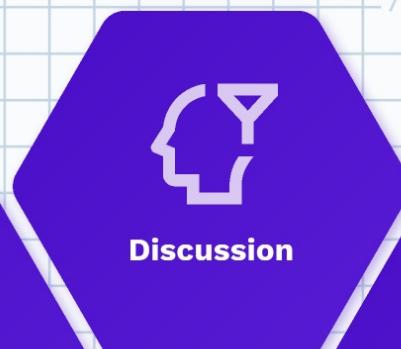
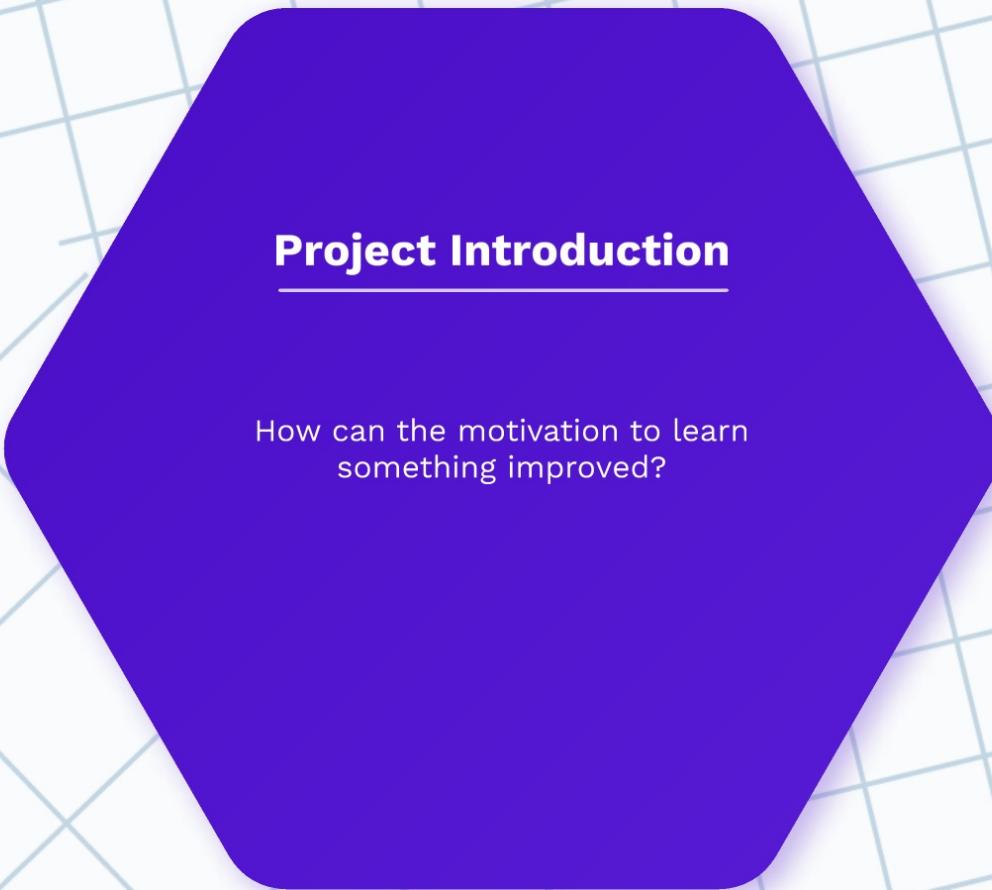


Quiz M&M Dispenser

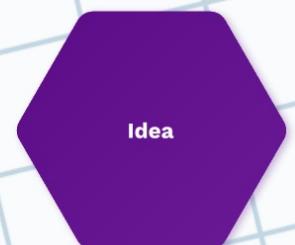
Martin Radinger
Nikolai Strässle
Pascal Steiner





Project Introduction

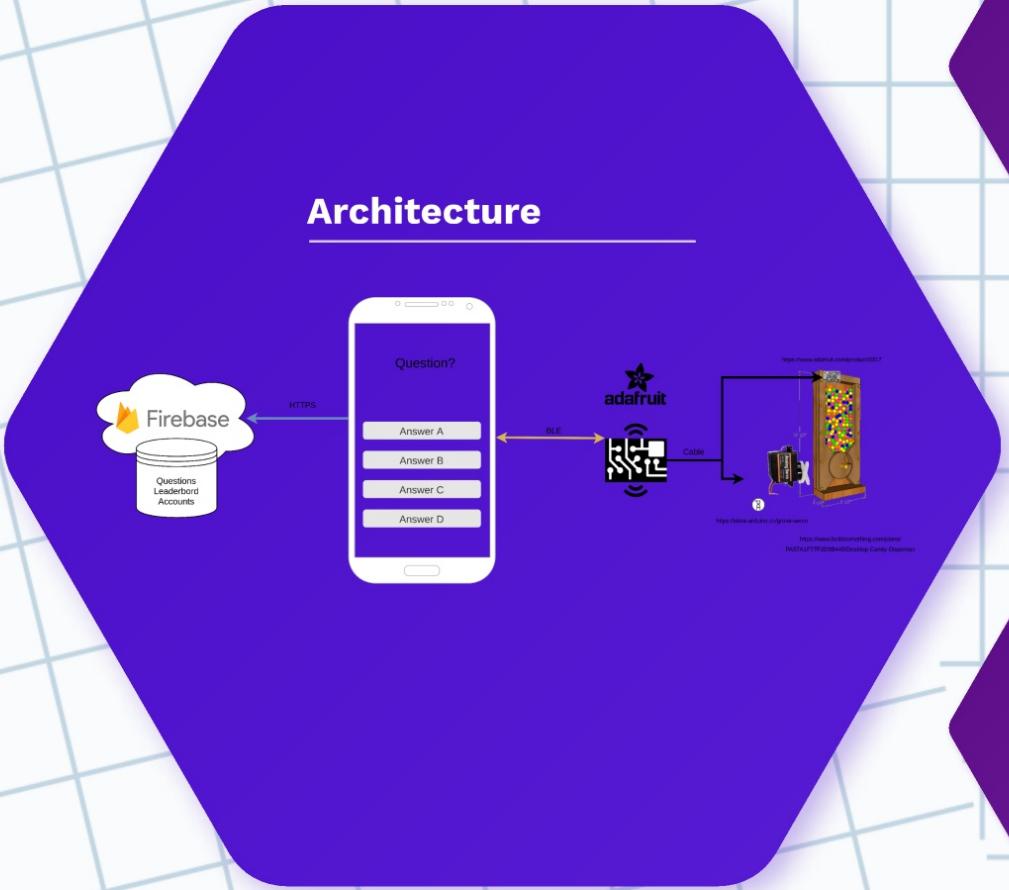
How can the motivation to learn something improved?



Idea

Goal:

- improve learning process in a playful way by giving incentive for learning and answering correct and gaining knowledge
- building the dispenser by ourselves

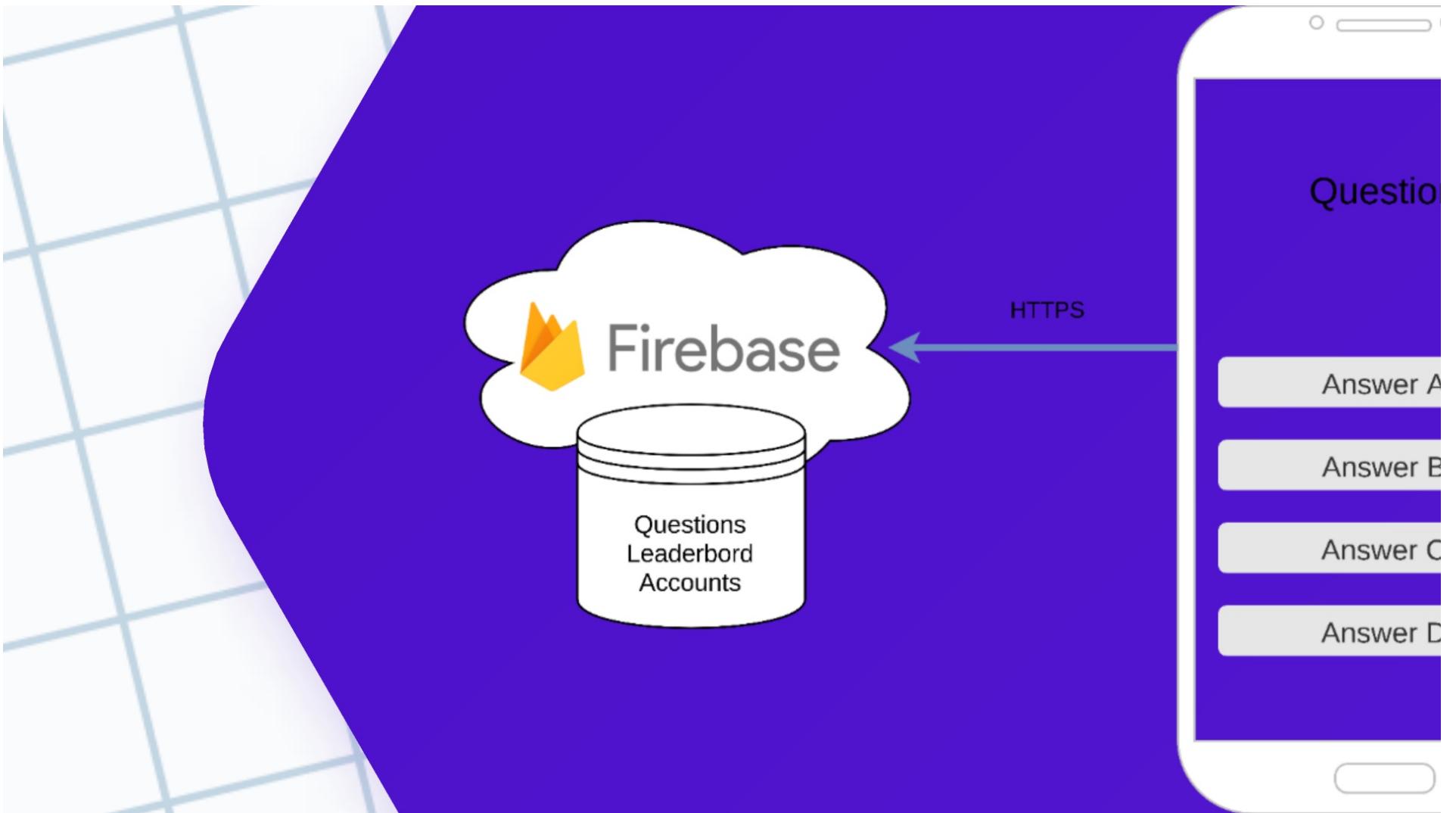


Firebase

BLE

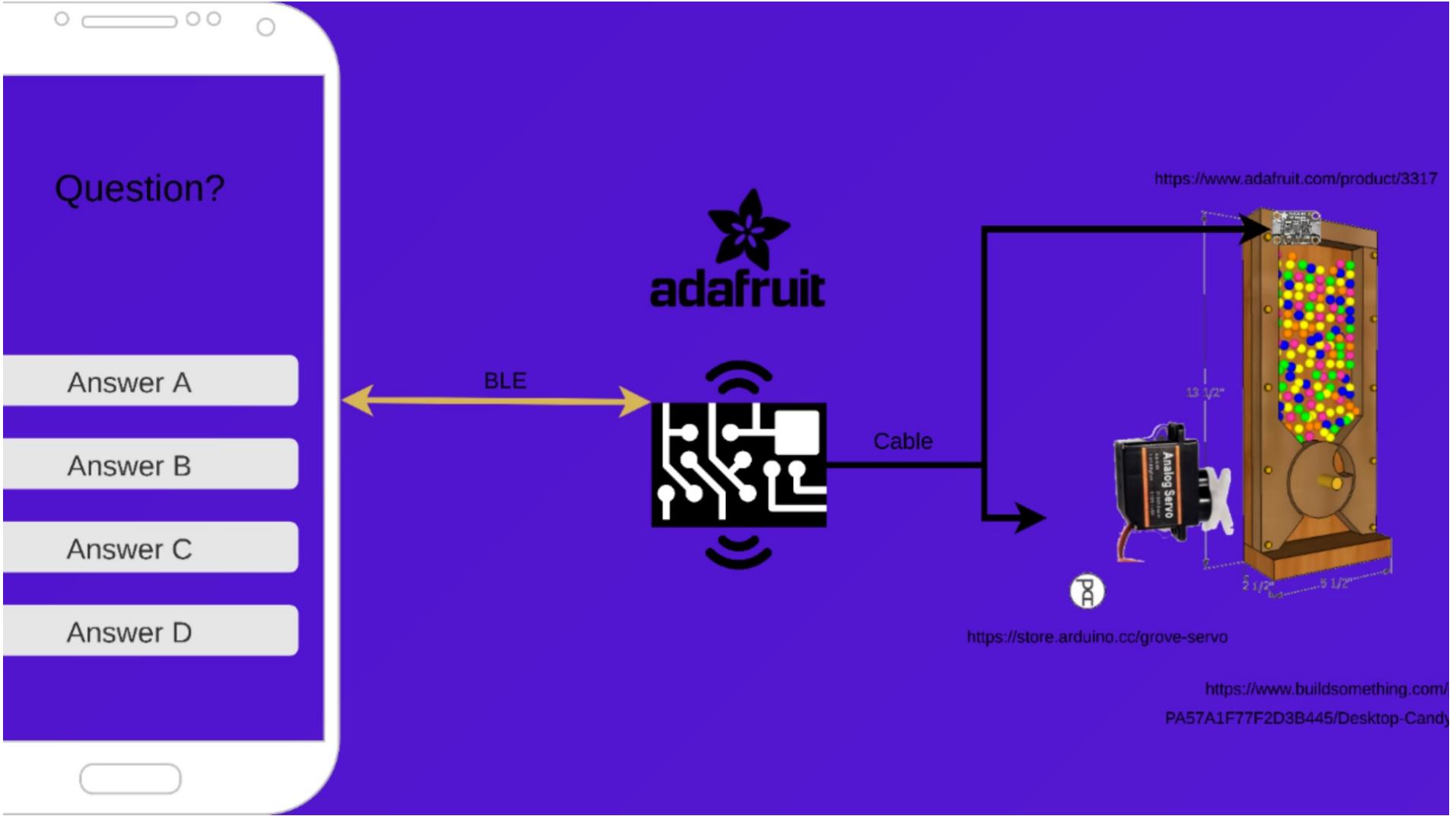
Sensor & Actuator

Our architecture and end-2-end solution. Interconnection of different components:
 firebase DB: real-time database to store data
 android app: answer questions to get rewarded, connected via ble to adafruit feather sense board, app as central and board as peripheral. App receives notifications from board and can send a dispence command



Firebase DB to store quiz data (questions & answers) and user data (login data, score).

- there are no security rules, you can create, update or delete database with API key
- limit of free requests per month



Custom GATT service, that has 3 characteristics

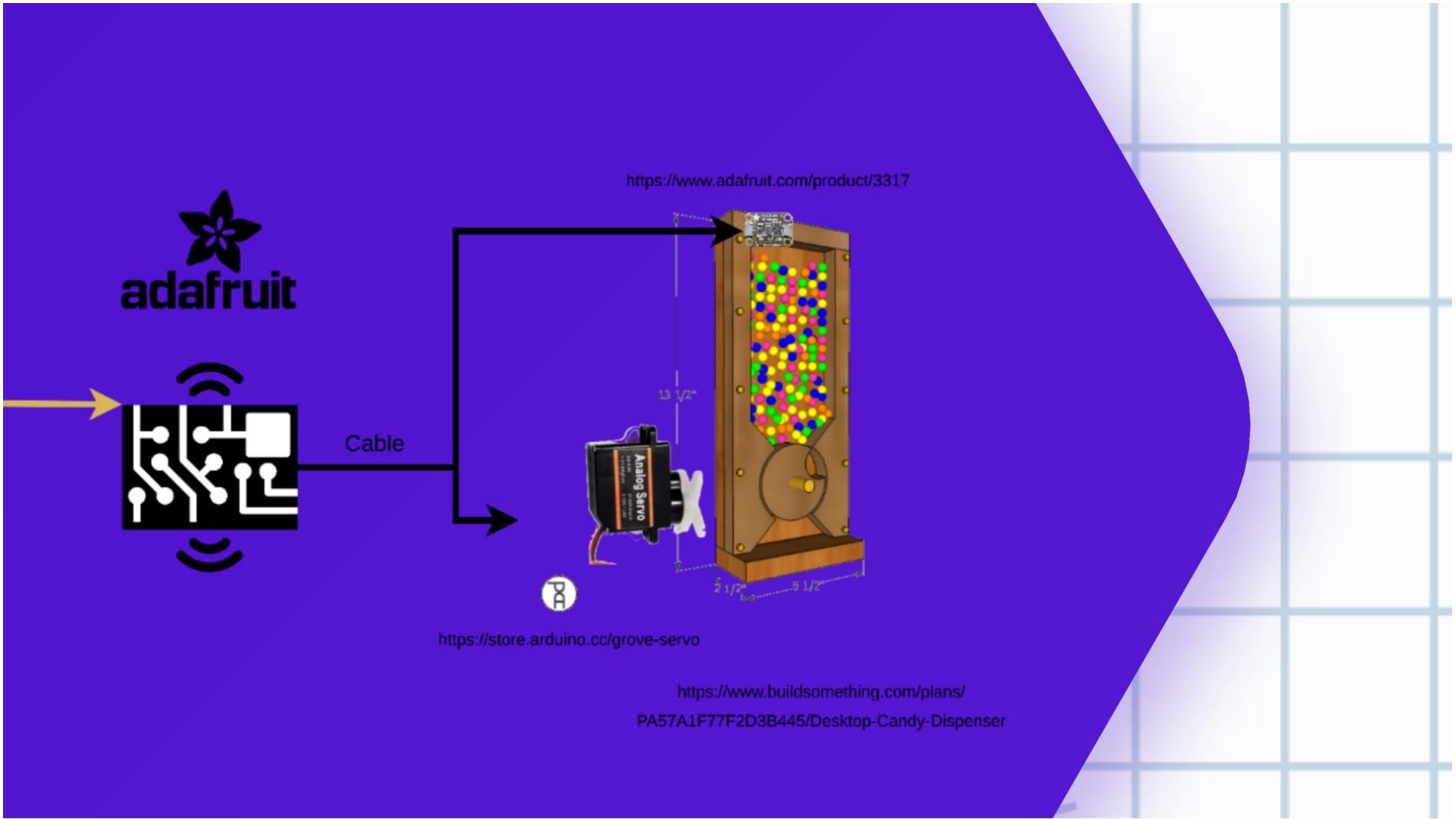
- #1 dispenser state characteristic notifies central about current state of servo engine (1: running, 0: waiting)
- #2 central can send dispense command (if central sends a 1 the board will start the servo)
- #3 sends current measurement of tof sensor to central

BLE

Custom GATT Service (113A0001-FD33-441B-9A57-E9F1C29633D3)

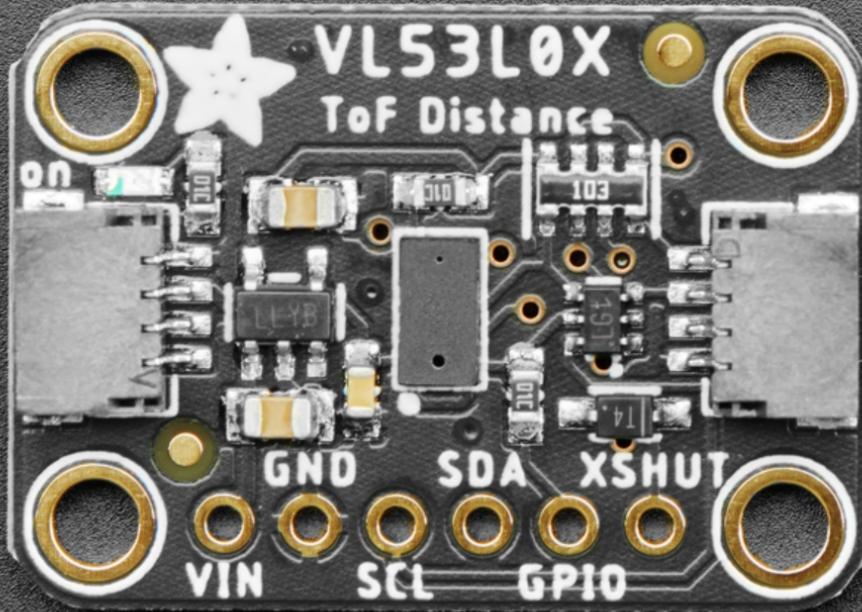
3 custom GATT Characteristics:

- Notify Dispenser State (113A0002-FD33-441B-9A57-E9F1C29633D3)
- Write Dispense (113A0003-FD33-441B-9A57-E9F1C29633D3)
- Notify Filling Level (113A0004-FD33-441B-9A57-E9F1C29633D3)



M&M Dispenser hardware:

- 1 actuator
- 1 sensor



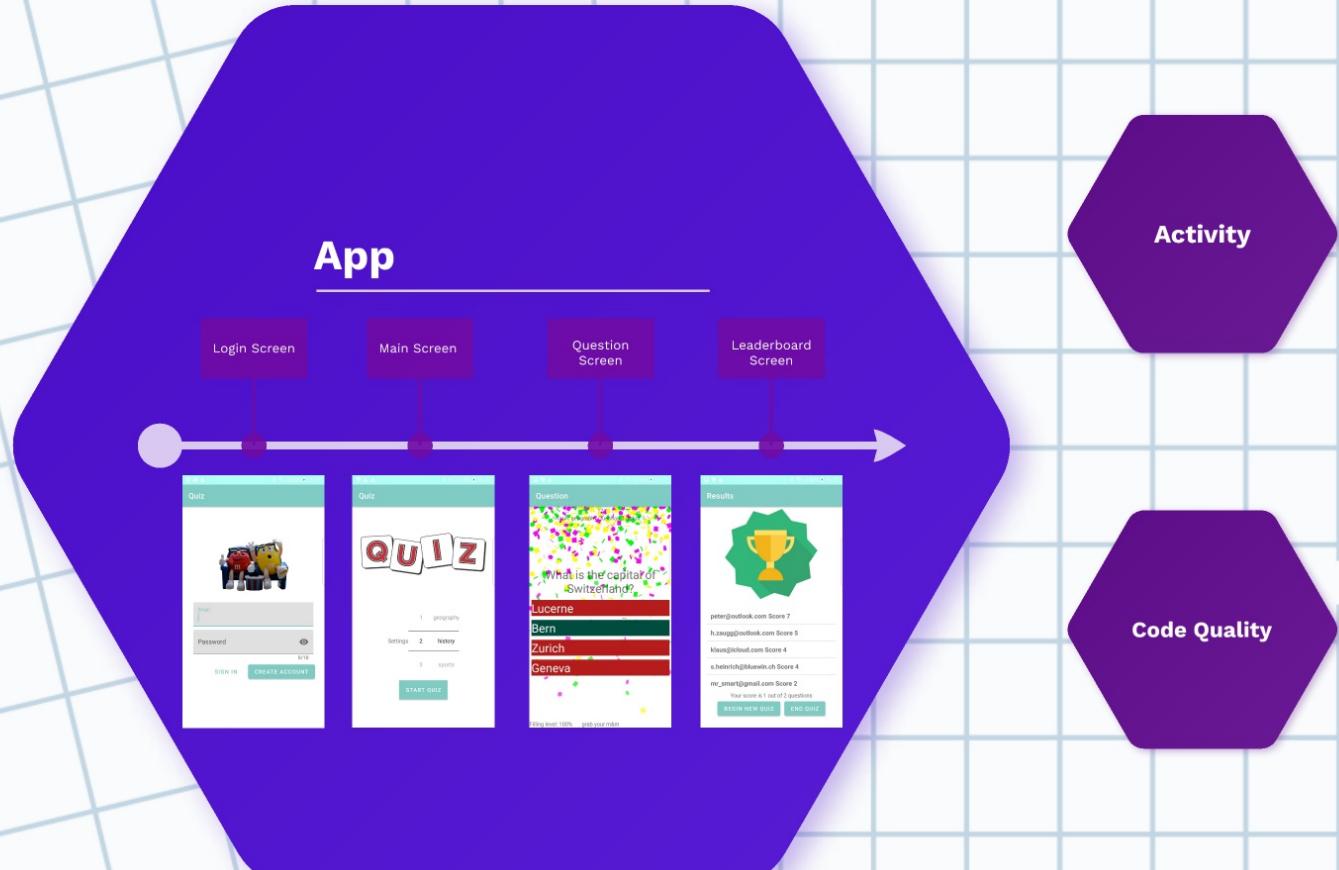
Adafruit time of light sensor to measure distance from lid of dispenser to the M&M inside the container.

To prevent measurement fails we average the last 5 measurements and send these to central.

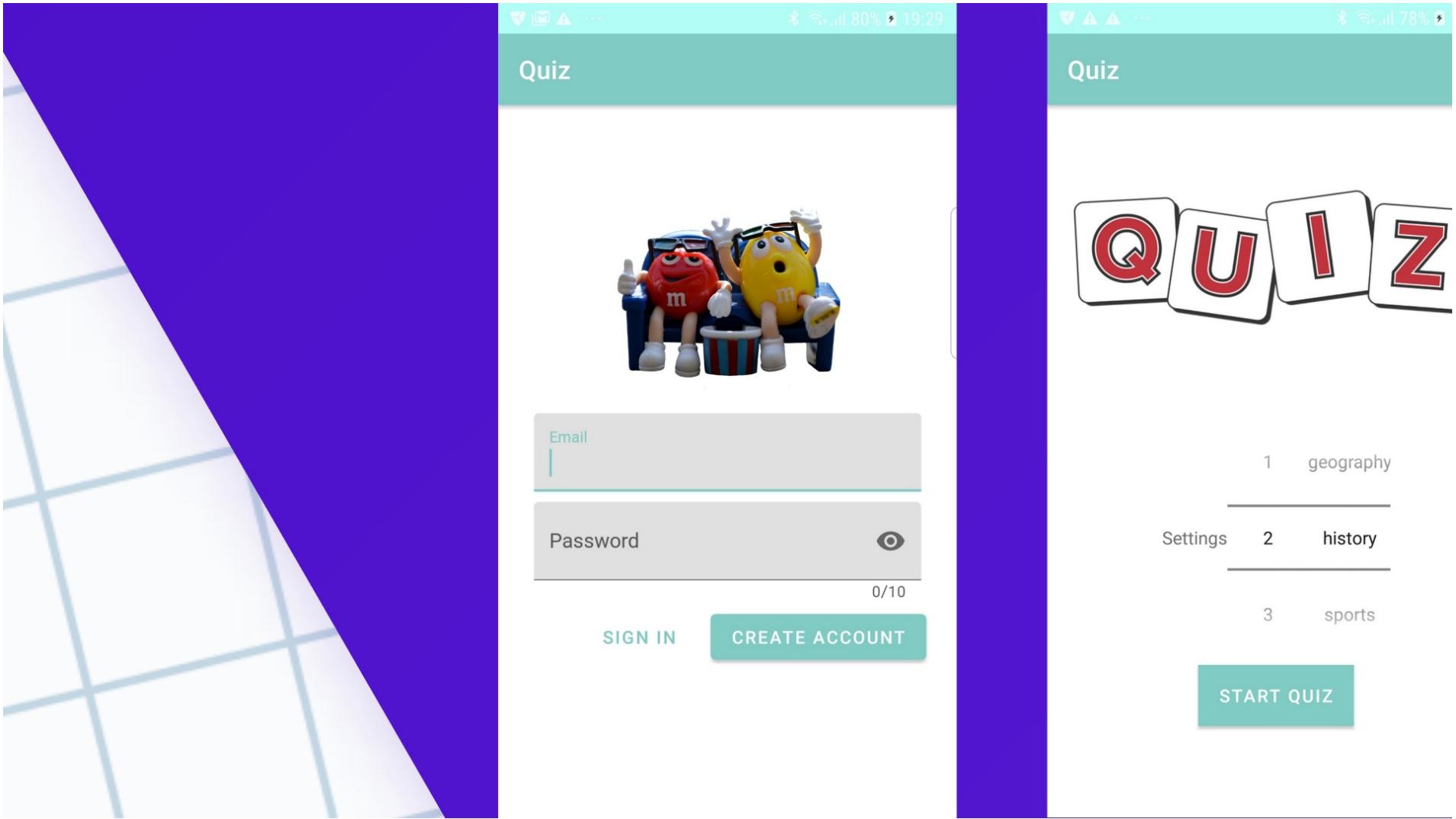
If sensor measures out of range, value is set to 2000mm and is an indicator that lid is open (someone stealing or refilling)



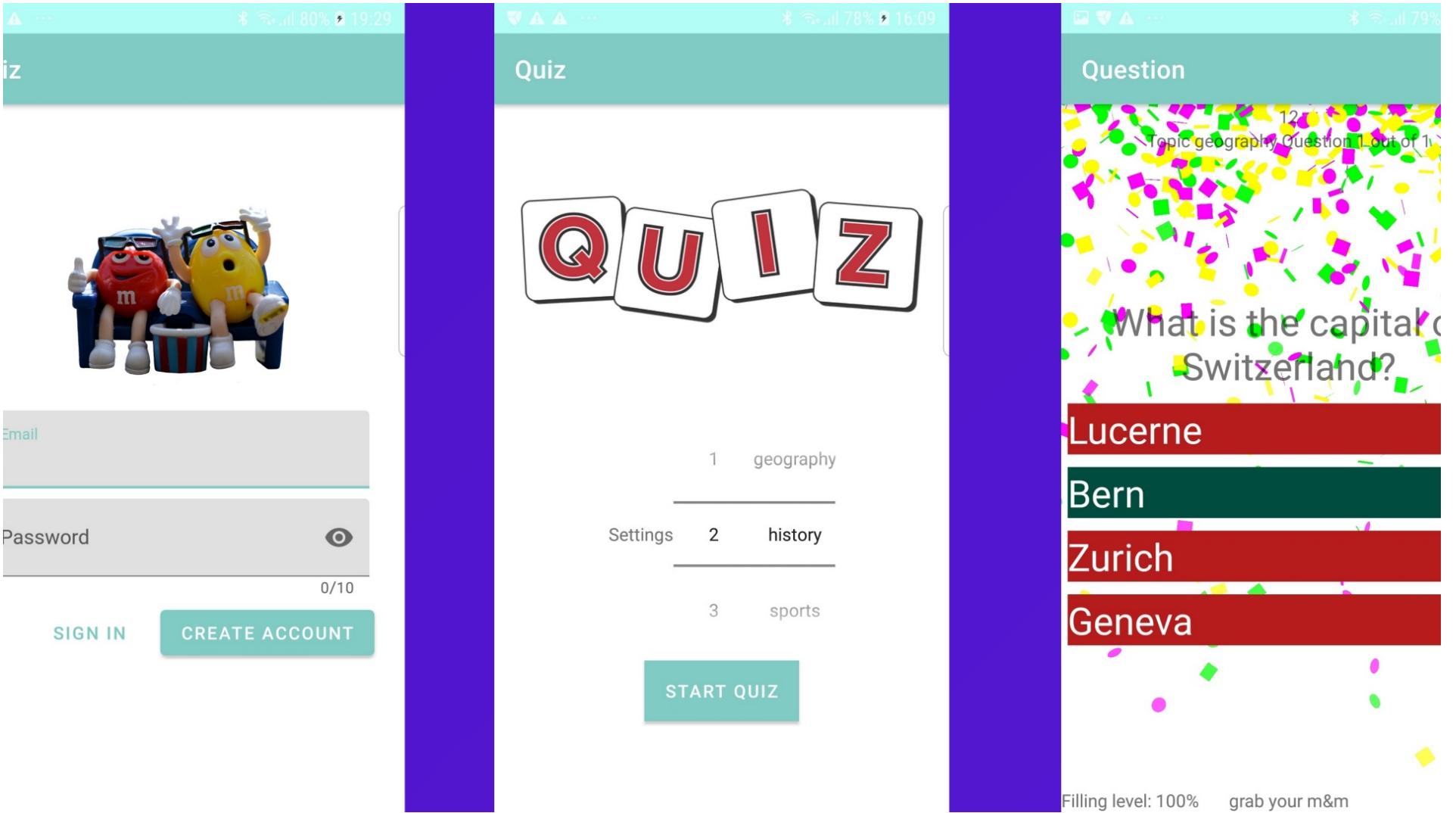
Arduino board is connected to servo engine which is attached to wood wheel of dispenser. If board receives dispense command from central via BLE, board starts the servo engine. First the engine rotates to the starting position and then rotate the open part of the wheel directly under the container. Then parts of the content will fall inside this open part and the engine rotates back to the starting position.



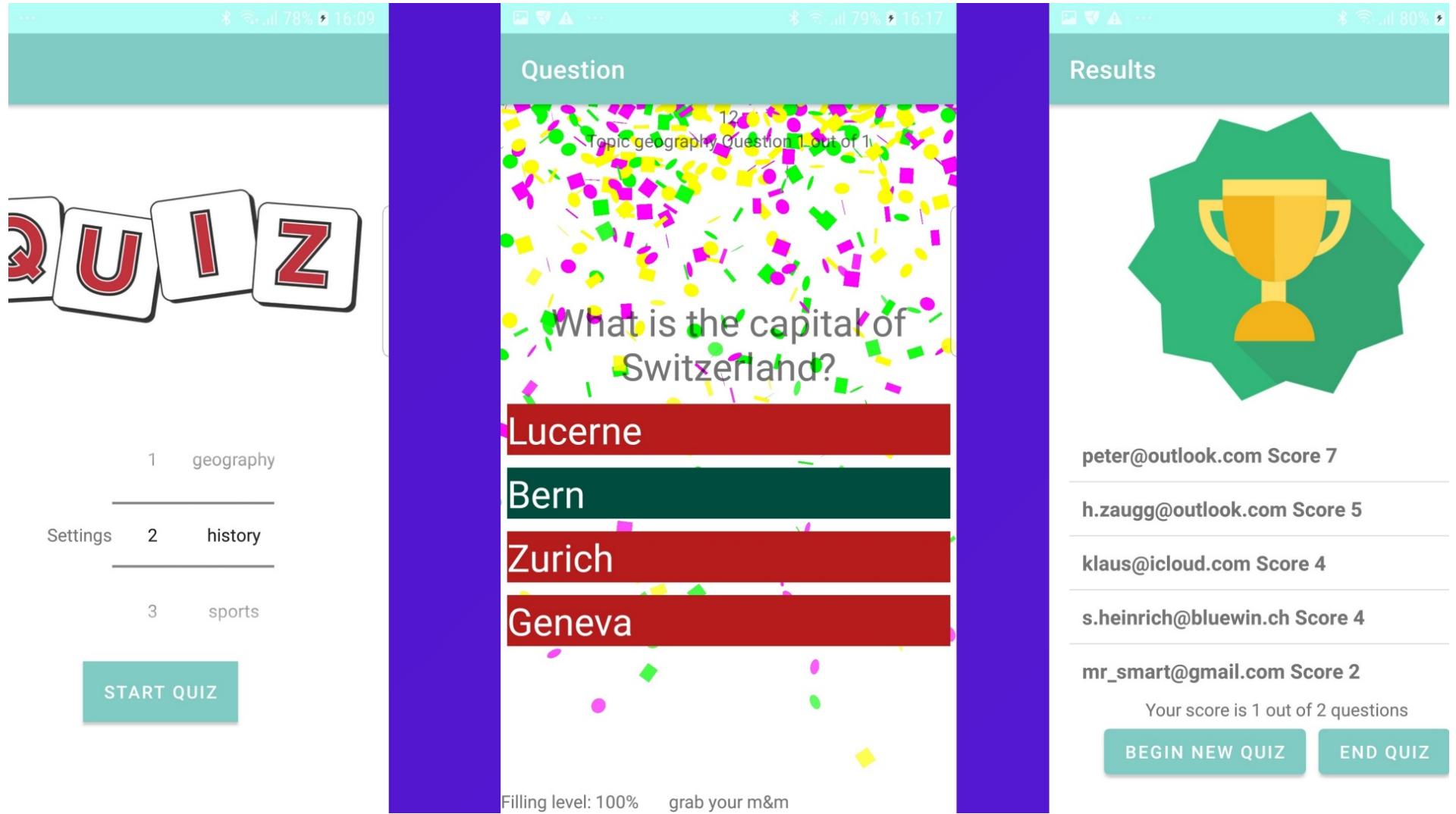
Images are all creative common license and free to use, color scheme based on android material design.
 Total 4 separate activities
 FirebaseLogin, MainActivity, QuestionActivity, QuizResultActivity



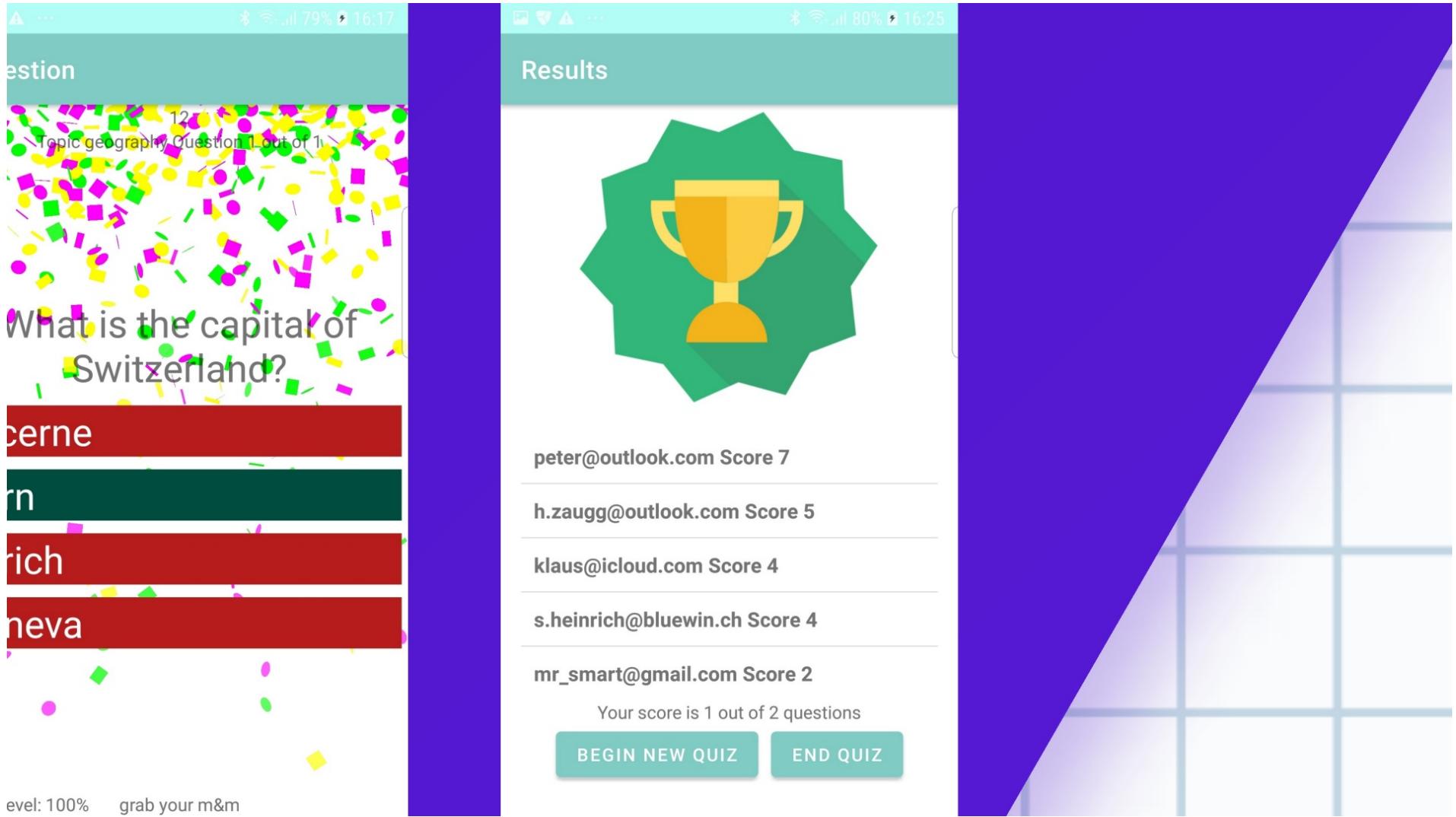
Firestore Login Activity: either create new account or sign in. Only e-mails in correct form and passwords with length>5 are accepted. Data is written / read from firestore realtime DB



After logging in, data from DB is read and available topics displayed in the numpicker. Player is only able to start the quiz once BLE connection is successfully connected via the BLEService, otherwise a toast is displayed. Connection is automatically established with correct uid of our BLE Service.

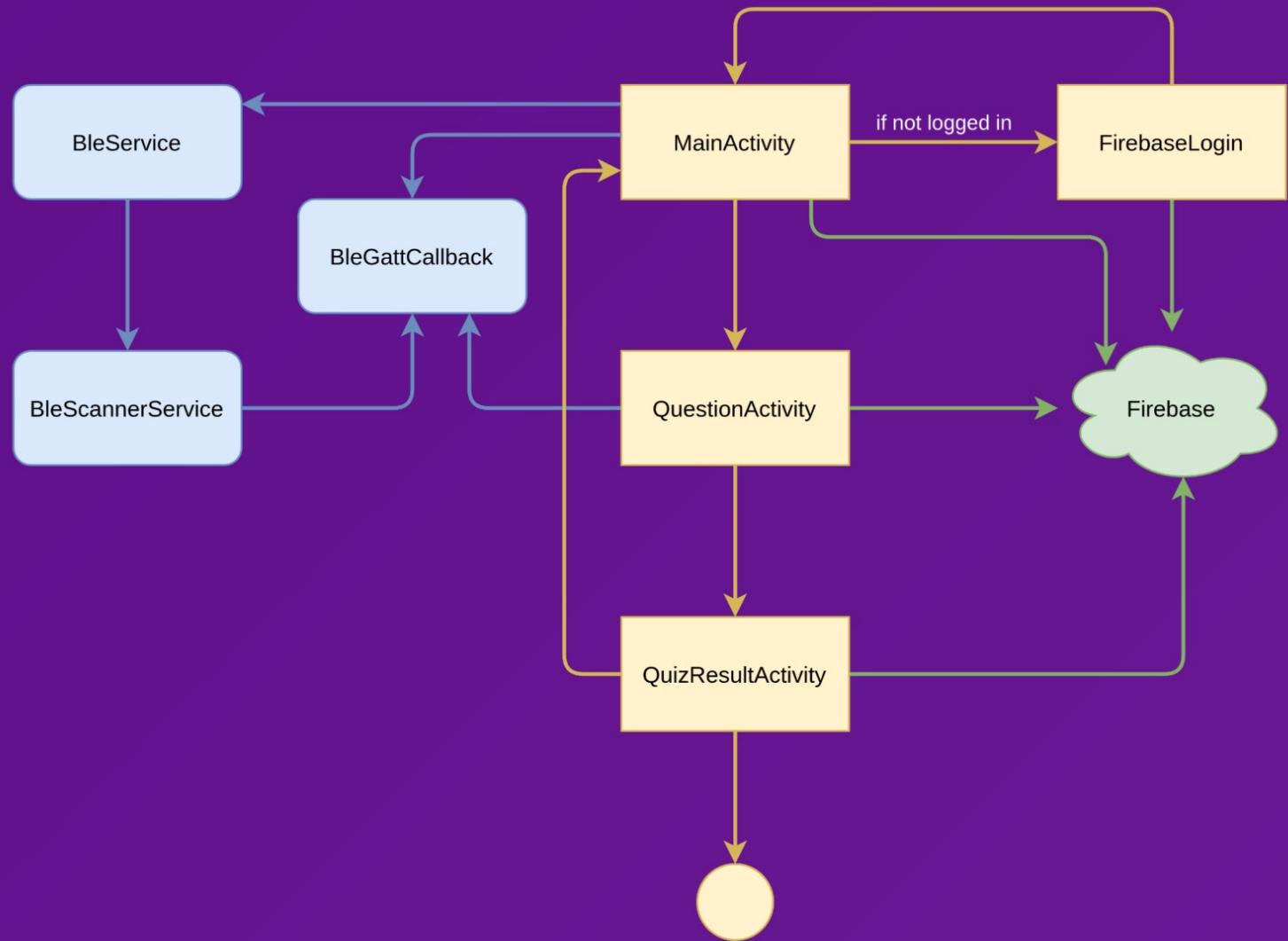


QuizActivity, handles the main part like reading all questions (randomly) from DB for the chosen topic.
Specifications: - timelimit per question 30 seconds, 4 multiple choice answers
When either time is up or all questions answered, QuizResultActivity is called.
Filling level and dispenserState shown at bottom, which is notified from the peripheral.



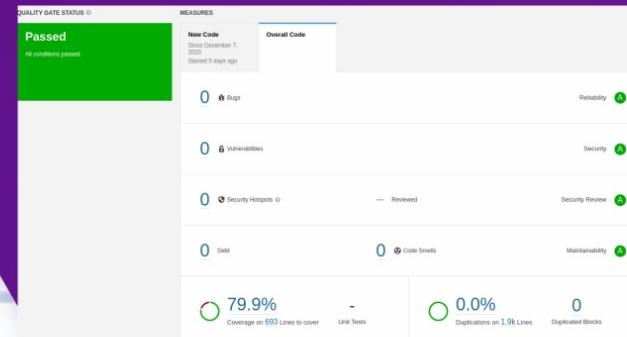
QuizResultActivity is last activity.

- writes user data to firebase DB and reads top 5 from leaderboard
- shows individual score of user and also the top5 overall for the topic
- user has choice to end quiz or start from beginning and play another round

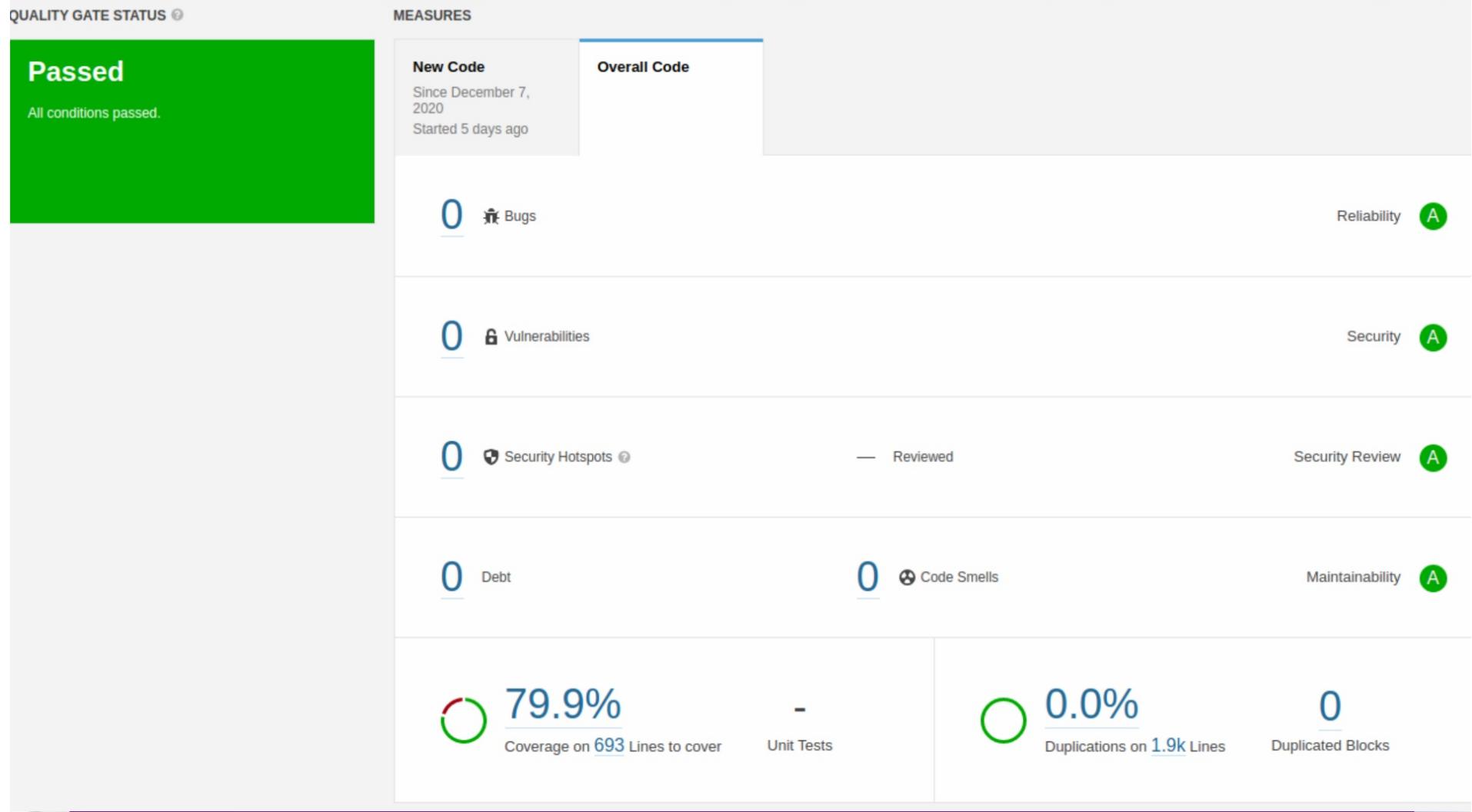


Code Quality

- 44 Unit Tests
- 8 End-2-End Tests
- 0 Security Issues
- 0 Bugs



Espresso for testing UI interactions
Mockito and robolectric for firebase and unit tests



In detail view from SonarQube output.

- we have significant code test coverage for a proof of concept project
- no duplications
- fixed code smells and security issues

Discussion



Achievements

- BLE Connection Android <-> Arduino
- Usage of sensors and actors
- writing unit and End-2-End tests

Problems

- Android Testing without experience is hard
- Debugging BLE connection was complicated
- Including sensors and actuator in a construction consumes a lot of time

Outlook

Summary and conclusion:

+BLE connection was challenging task and combining with sensors and actuators

+Test coverage

Problems:

- writing tests was harder and took longer than expected

- debugging BLE connection: difficult to keep track on what has been sent and current state

Outlook



- Improve the dispenser construction
(e.g. hide electronic parts)
- Add a duel feature
- iOS application
- Automatic reorder new M&M's if the filling level is lower than 10%

Prototype

