Development Testing:

Throughout the development phase of our app we will use a variety of tests to ensure all features function as intended. The two main types of tests used during development will be ‘Unit tests’ and ‘Widget tests’. Unit tests operate on a single method, function, or class and are used to check logic operations are computed as intended. This will often involve testing the method, function, or class under a variety of conditions. Widget tests are slightly more thorough and aim to test that a widget both functions and appears on screen as planned. This testing process makes use of a testing environment and several classes to provide a situation where a widget can be tested.

Testing our scheduling algorithm:

We’ve chosen this section of our app to talk about since it’s fundamental to providing the correct pieces for students to review. Testing it thoroughly will ensure it functions as intended and doesn’t fail as this could negatively impact students' learning and progress e.g if a piece far too easy or difficult is scheduled it would be a waste of the students learning time.

Test Case
A piece is only scheduled for review during that week if the number of pieces due for review is below the threshold.
All pieces in the “Learning” section stay the same until the teacher marks them as “learnt”.
Suspended pieces are never due for review.
Unsuspended pieces are pushed to the back of the review queue.
No pieces are in the “Learning” or “Reviewing” section if learning and reviewing is paused respectively.
A piece marked as “hard” is scheduled to the next available review slot at least a week away.
A piece marked as “easy” is scheduled to the next available review slot at least 3 weeks away.

Release Testing:

For high-level release testing we will use a testing method involving 'Integration tests'. These tests work by testing the entire app or very large sections of it and aim to see whether components and widgets are interacting with each other as intended. These tests can also be used for performance testing/ optimisation of the app. Typically these tests are run on a real device or emulator and are often run manually.

Testing a user story:

"*As a* ***piano teacher****, I want to be able to* ***monitor all my students' accounts*** *so that I can* ***customise each of their routines to fit their calibre***".

We've chosen this user story to talk about since it is a key part of the app's overall intended purpose. Ensuring a teacher can monitor all their students and make changes to their practice plans is essential.

Test:	**Intended outcome:**
As a teacher, add a piece to a students profile for review.	The piece appears on the student's account as one they are required to practice.
As a teacher, remove a piece from a students profile.	The piece is removed from the student's list of practice pieces.
As a teacher, click on a students profile.	An updated student profile is displayed showing current set practice pieces and other student info.
As a student, link to a teacher using a unique teacher code.	Student is added under the teacher's list of students and the teacher is added to the student's account.
As a teacher, be able to search for a student in a list of all students.	A search for a student's name or unique id should find them in a list.