Suzuki Practice Buddy

An application that helps children learn to play the piano and improve their technical skills by making practising more enjoyable.

Joe Harrison, Thomas Plumtree, Yuqing Zhu, Aadith Pillai, Daniel Clarke

Overview

Summary

Client: Anna Sibley

Organisation: Bristol Music School

Anna Sibley is the founder and head tutor at Bristol Music School. She has been teaching the Suzuki Method since 2010 and has been very successful in inspiring many young pianists. However, due to their short attention spans, she has found it hard to get her students to engage with their practice even for 5 minutes. Seeking a solution to this problem, she approached us with an idea for an app that will help her younger students improve their skills while also having fun.

Our app transforms the seemingly dull routine of practising into a fun activity that the students look forward to. It acts as a checklist that will reward the student each day based on how much they complete. The tasks completed each day are recorded and there is an indicator showing which tasks have been completed each day of the week, encouraging the student to practise consistently.

The selection of pieces is in line with the Suzuki method, which is a teaching method where a student progresses linearly through a set of pieces and books, while also working on previous pieces based on the same technical skills as the piece they are currently learning. The app is intended for use by the students working on the first 3 books. These students will be aged between 4-8. The teachers may monitor a student's progress on the student's device to see what they have done between lessons or to move them on to the next piece if they have made enough progress.

Features

Tasks

On any given day, the student has the following tasks to complete:

- **Learn:** The student needs to practise the piece they are currently learning. They start from the first piece in book 1 and progress linearly across the 8 books. Only the first 3 books are on the app as it is only aimed at younger students.

- **Review:** The student spins a “wheel of fortune” to pick a piece to review. The pieces on the wheel depend on the current piece the student is learning. The student can spin the wheel as many times as they want. Each time they review a piece it is removed from the wheel until the next day. If they review all the pieces on the wheel, the wheel is reset.
- **Listen:** The student has to listen to their recording (separate to the app) and then mark on the app that they have done so.

Rewards

Each day, the student earns badges when completing a task. They get a bronze badge if they complete one of the tasks, silver if they complete two, and gold if they complete all 3. Students also receive coins after completing a task which they can spend to buy new backgrounds in the store. They can then apply these backgrounds to make their home screen more exciting.

Students can also complete achievements for doing certain things, including earning a certain number of coins or completing a task a certain number of times.

Requirements

Stakeholders

- **Anna Sibley:** Since the idea for the project was given by Anna Sibley, the founder of Bristol Music School, she has full control over the development of the project and will want her students to use the app to make their piano practice more enjoyable.

- **Piano teachers:** This app is intended to encourage students to do their piano practice, so it should allow the teacher to make their students’ practise more fun and engaging, which in turn will make them practise more and improve as pianists.

- **Students:** The students will be using this app to encourage them to do their daily piano practice while also making it fun and rewarding.

- **Parents:** As the app is aimed at children, the parents would likely need to download the app on their phone so their child can use it. They will also want their children to do their practice properly and will be there to supervise them during each practice session.

- **Google Play Store:** This app will be published on the Google Play Store, so it will need to conform to all the required rules and regulations.

Interactors

The main interactors with the system will be the **students**, as they will be the ones encouraged to use the app to make their practice more entertaining. **Teachers** will also have a keen interest in their students' use of the app as they will want them to practise to improve their skills and the app acts as an incentive for the students to practise. The **parents** of the students will also have an interest in the app, as they will want their children to get the most out of their lessons and practice. As the children the app is aimed at are quite young, it is also likely that the app will be installed on the parents' phones as the students may not have their own phone to use the app.

Students

- ***"****As a* ***student****, I want to* ***have an incentive to practise*** *so that I can* ***maximise the productivity of my practice sessions.****"*

Our solution: At the end of every review, the student is rewarded with coins which they will be able to use in the store to buy new backgrounds. Backgrounds can then be applied to make the homepage of the app more exciting. There are also achievements that the students can earn if they complete a certain number of tasks or buy a certain number of items. Badges are awarded each day if the student completes their current piece, a review piece, and a listening task (gold for all three, silver for any two and bronze for only one).

- ***"****As a* ***student****, I want to* ***review the same piece multiple times*** *so that I can* ***improve my performance.****"*

Our solution: When the student has reviewed all the pieces on their review wheel, the wheel resets so that they can go through all the pieces again. Each piece due for review is, therefore, able to be reviewed any number of times before the next lesson. The wheel also resets every day, so the students can start their practice from a clean slate each day.

- ***"****As a* ***student****, I want to* ***record the practice I have done today*** *so that I* ***know what I still need to do.****"*

Our solution: When the student marks their daily practice of their current piece as done, the app tells them it has already been practised. The same happens with the listening task.

Review pieces are removed from the wheel when they have been practised. This way, the student cannot spin the wheel and land on the same piece twice, unless they have gone through every piece on the wheel, in which case all the pieces are put back on the wheel.

Piano teachers

- **"***As a* ***piano teacher****, I want to be able to view* ***my students' progress*** *so that I can* ***understand where they are at*** *and* ***set their new goals accordingly.***"

Our solution: The student records the tasks they complete on the app on their parents' phone. In the lesson, the teacher can look at what the student has done since their last lesson by viewing their badges. The teacher, if they feel the student is ready, can move the student up to the next piece in the book.

- **"***As a* ***piano teacher****, I want my students to be able to* ***pick random pieces to review*** *from the list of review pieces so that they* ***do not pick the same pieces each time.***"

Our solution: Students can spin a "Wheel of Fortune" populated with the names of pieces on the review list of their current piece. They will then practise the piece this wheel lands on and be able to mark this on the app, then spin the wheel again to pick another piece to review.

Each time a piece is reviewed, it is removed from the wheel until it resets either the next day or if the student has reviewed all the pieces.

- **"***As a* ***piano teacher****, I want to be able to* ***make my students' daily practise as fun as possible*** *so that they are encouraged to* ***improve and keep learning.***"

Our solution: The app will provide different ways for the student to engage with their practice and have more fun than if they were just doing it out of a book, including:

- A review piece "Wheel of Fortune",
- Gold, silver, or bronze badges each day depending on how much practise the student has done,
- Coin rewards for doing tasks that they can use to customise their app to make it more interesting,
- Achievements that the students can earn if they complete certain elements of their practice consistently.

Parents

- **"***As a* ***parent,*** *I want my child to* ***engage in their piano practice*** *so that they* ***make the most of their lessons.***"

Our solution: The app intends to make practising more entertaining for students so that they want to practise more. The features mentioned above will make practice feel more like a game so the children will want to do it more often and will therefore develop their skills quicker.

System Flows

The main function of the app is for the student to track their daily practice while having a bit of fun at the same time. The way a student might navigate through the app could be as follows:

Normal Flow

1. The student opens the home page.
    a. The student marks the piece as done and receives a reward.
2. The student opens the listening page.
    a. The student marks listening as done and receives a reward.
3. The student opens the review page.
    a. The student spins the wheel.
    b. The student reviews a piece on the wheel and receives a reward.
    c. The student can optionally repeat step 3.
4. The student opens the shop.
    a. The student purchases an item.
    b. The student applies the new item.

Alternative Flow

1. The student opens the shop.
    a. The student tries to purchase an item.
    b. The student is told that they do not have enough coins.
2. The student follows steps in the **normal flow.**

Exceptional Flow

As there is no data to be input, just buttons to press, there is no exceptional flow on the app. Users are only able to open the 3 main pages and the shop and then carry out the desired action without inputting any data.

Atomic Implementation Features

Home Page

- The home page shows the user’s current piece.
- The user can mark their current piece as done no more than once a day.
- The home page has links to:
    - The review page, where the user can spin the wheel to select the review piece.
    - The listening page, where the user can mark their listening as done no more than once a day.
    - The shop, where the user can buy items.

    - The inventory, where the user can access their purchased items.
    - Settings, where the user can access their achievements, the tutorial, and reset their progress.
- The user can tap on the corresponding navigation button and be navigated to the corresponding page.
- The user can change their current piece by clicking on it.

Review Page

- The user can spin the wheel if they have learned more than one piece.
- The wheel is populated with the pieces in the review list of the current piece. This list has been decided in advance for every piece, as part of the Suzuki Method.
- The wheel is colourful, with each segment (each piece) coloured differently from the ones around it.
- When the wheel is spun, the piece it lands on is shown to the user.
- The user can mark the selected piece as reviewed.
- This piece is removed from the wheel when reviewed.
- When there is only one piece, the piece is shown instead of the wheel.
- If all pieces have been reviewed, the wheel resets.
- At the start of every day, the wheel resets.

Shop

- The shop displays backgrounds that the user can purchase.
- If the user clicks on a background, they are given the option to purchase it.
- If they have enough coins, they can then purchase the background.

Inventory

- The user can access their purchased items.
- The user can apply one of their backgrounds by clicking on it in their items list.

Personal Data, Privacy, Security and Ethics Management

The app does not collect any personal data about the users and there are no accounts. There are also no cookies being stored. The app does not require a connection to the internet for it to work and all data is stored locally in a database on the users' device. The data that is stored is related to the pieces in the books, whether the user has done each element of their practice each day, coins, and the backgrounds for the shop. As such, we do not have any security strategy as the data is unlikely to be of any interest to potential hackers. There are also no ethical concerns that we need to consider safeguarding the users. That said, the users themselves are responsible for their data and can lock their phone if they feel the need to do so. **Ethics approval was applied for on 24/11/20 at 21:41.**

Development Testing

Throughout the development phase of our app, we used a variety of tests to ensure all functions and features work as intended. The two main types of tests used during development were ‘Unit tests’ and ‘Widget tests’. Unit tests operate on a single method or function and are used to check logic operations are computed as intended. They are also deployed when testing various database functions responsible for retrieving and processing data from a database. This involves testing the method or function under a variety of conditions and ensuring the correct data/result is returned. Widget tests aim to check that a widget appears on the screen as intended. To do this we made use of ‘flutter_test’ which is a library built on top of the ‘test’ package in Flutter. It works by building a given widget using a set of parameters. A ‘finder’ is then used to search a widget tree and return nodes that match a particular pattern. For example, testing to see whether a text widget appears in the given widget tree and comparing this to a ‘matcher’ to see if the widget is displaying the correct text.

A core feature of our application tested using these techniques is the review page, which includes the review wheel as well as a preview of the piece currently selected by the wheel. We have chosen to analyse the test cases used for this component since both unit and widget tests are required to ensure it is functioning as intended. The review page is an essential part of the application since it is responsible for providing an engaging and interactive random piece selection process. It is also important that the pieces appearing on the wheel are the correct ones so that a student does not waste time practising incorrect pieces.

Test action	Intended outcome
A **piece** is added to the **review list**. (Unit test)	The **correct** piece is added and stored in the **review pieces table** on the **database**.
The **review list** is requested. (Unit test)	The **entire** review list is fetched from the database and returned. It contains all the **correct** pieces.
The **piece preview** widget is built. (Widget test)	The piece preview displays the **correct** piece name.
The **review wheel** is built using the **review list**. (Widget test)	The wheel displays **all** the **correct** piece names from the given **review list**.

Release Testing

Our approach to release testing our application consisted of mainly manual testing conducted by members of the team. This process involved deploying the application in debugging mode to either our smartphone or an emulator using ‘flutter run’. From here we were able to thoroughly test components by interacting with them under different conditions to ensure they performed as intended. Any errors were printed in the flutter console and other unintended behaviour was easily spotted and rectified.

Testing a user story

"As a* student*, I want to* have an incentive to practise *so that I can* maximise the productivity of my practice sessions*."

We have chosen to analyze the test cases relating to this user story since designing a fun and interactive learning incentive was one of the client's key criteria. Keeping young children engaged in their piano practice can be challenging, so it is important that they enjoy the learning process.

Our solution to this problem was to design a coin system and item shop where students can purchase decorative backgrounds that can be applied to their home screen. This system must function as intended so that students remain motivated and interested.

Test action	Intended outcome
An **affordable** background is purchased from the shop.	The background item is marked as **sold** in the shop and is added to the user's inventory. The background price is deducted from the user's coin value.
An **unaffordable** background is purchased from the shop.	A message is displayed informing the user they do not have enough coins for the background. The item is **not** marked as sold and **does not** appear in the user's inventory. **No** money is deducted from the user's coin value.
A **background** from the user's inventory is **applied**.	The user's home screen background is **updated** to the **applied** background.
A user completes their **first, second, or third** task of the day.	The badge promotion animation appears, awarding them with a **bronze, silver, or gold** badge for that day, depending on how many tasks have been completed. **100 coins** are added to their coin total.
A user completes another task after having earned **all** their badges for that day.	The regular coins animation appears and **100** coins are added to their coin total.
A background marked as **sold** in the shop is pressed.	The user has the option to **apply** the background rather than **purchase** it.

Architecture

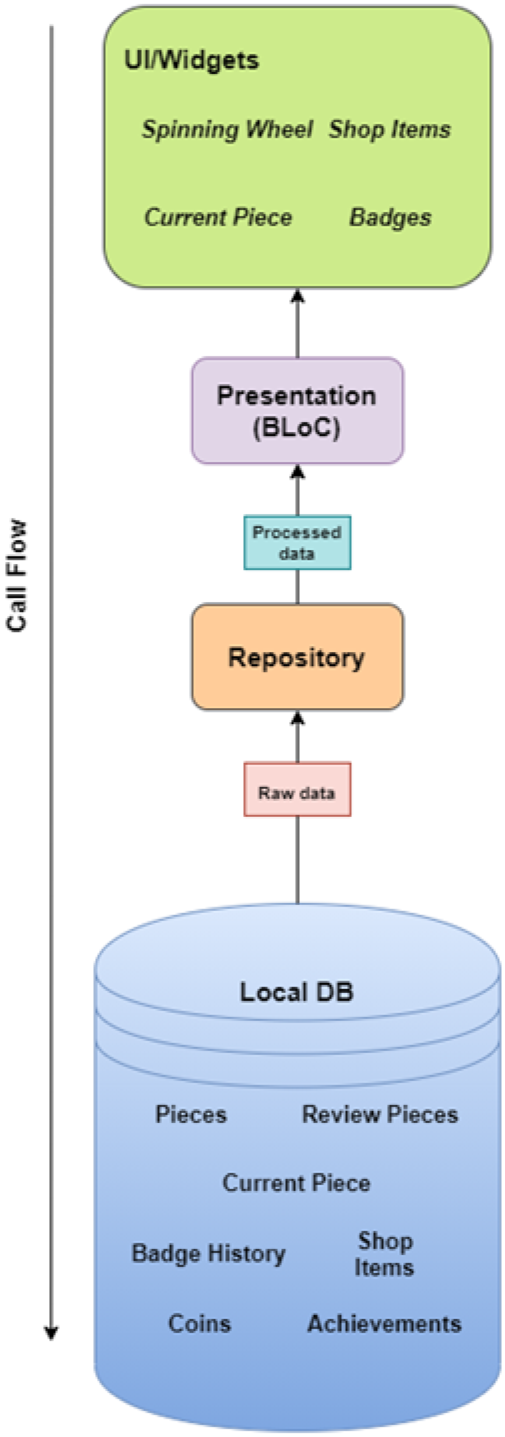
We used Flutter and Dart to build our app. We also made use of Dart libraries like async to enable asynchronous programming, and flutter_test to write tests. All data is stored in the phone locally so an Internet connection is not required. It is made up of the following components:

UI/Widgets: This is the part of the system that appears on the screen and the user directly interacts with. Examples of such elements include the spinning wheel, shop items, and badges.

Presentation: The widgets in the previous layer dispatch events to the BLoC (Business Logic Components) and listen for states. Widgets submit events, and the other widgets respond. This is especially helpful when communicating the number of coins between screens and updating the state of shop items (yet to purchase or purchased).

Repository: The repository contains all of the functions that the presentation layer needs. It receives data from the database and converts it into data that the presentation layer can use.

Local DB: This is the local database within which all information regarding the progress of the user is stored. It also contains static information such as the pieces and mappings of review pieces to the pieces themselves.



Why Flutter?

In our project, we use Flutter and Dart to develop the mobile application. Our app is used by children and applied on the phones of their parents. Some of them are using Android phones and others are using iOS phones so our app should both work on Android and Apple devices. Flutter is a cross-platform framework and it does not need any platform-specific UI components to render its UI. Also, Dart allows us to use the same codebase to build Android and iOS apps. This will enable us to effortlessly release the same application on the App Store if needed.

Why offline?

Once downloaded, our application does not need any internet connection and all information is stored locally. The core functionality of the application does not need the Internet or any external servers to store information as there is little benefit in doing so. All the pieces needed were sent from the client and hard-coded into our app. If there is a need to change the database that comes with the application, we can simply change the database and deploy it once again through the Google Play Store.

Why this UI design?

Our app is mainly used by children around 4-8 years old. Its objective is to make children interested in playing the piano without making it a competitive or stressful activity. Our UI design represents this. The animations on the different pages and the spinning wheel on the review page are designed to appeal to children. The badges and coins given after the completion of any task are visual rewards with the sole purpose of motivating them to practise. Items in the shop and achievements add another layer of incentive by giving them a goal to work towards. All of these features are shown in widgets and database parts of the diagram.

Static UML

The objective of the application is to make the learning process fun and interactive. Practising consists of learning to play a new piece and simultaneously regularly reviewing pieces learned in the past. To make this activity enjoyable, we decided to incorporate a spinning wheel. This wheel contains all the pieces that the student has already learned, narrowed down to those that will help the student practise the techniques required to learn the piece they are currently learning. They can then spin this wheel and review the selected piece.

This aspect of the application represents its core interactive nature. It turns the dull task of flipping through a book and picking a piece into a colourful, engaging activity. The spinning wheel was a surprisingly complicated part of the system to build. It communicates with the other pages and the database to generate the wheel and uses various circular motion equations for the spinning animation. It was especially difficult to design because we had to translate the coordinates provided when the user touches the screen to the axes that the animation uses. To account for this while making the code easier to debug, we decided to split up the functions related to motion into different classes and generate the wheel separately before animating it. This gave us more control over the elements such as the text on the wheel and the colours of the sectors. Furthermore, it made the wheel easier to animate, since each class was performing just one task. The classes shown in the figure have different roles:

- **Wheel:** This class takes a list and builds the wheel that is displayed. It starts with generating the colourful segments and then generates the text that is shown above.

- **NonUniformCircularMotion:** This class calculates the distance covered by the wheel while spinning and the duration of the animation using basic motion equations. The acceleration is based on a chosen resistance. This resistance can be adjusted to make the wheel spin faster, but it is a fixed value.

- **SpinVelocity:** When the user touches the wheel, Flutter provides the velocity in the form of “pixels per second”. To calculate the initial velocity, this class converts this value to one that uses x and y coordinates. It also converts these coordinates to radians, which is needed before applying the equations of motion.

- **SpinningWheel:** This class is responsible for animating the wheel. It takes the generated wheel, changes global coordinates to local coordinates relative to the position of the wheel, and uses the classes above to animate it. It waits for the wheel to stop spinning and calls a function “onEnd” to notify the parent class of the chosen piece (ReviewPage).

- **ReviewPage:** All of the widgets displayed on the screen are built here. It updates the database once the review piece is chosen and displays the promotion animation if this is the first review piece of the day.

- **Promotion:** The promotion animation is built here.

- **Piece/PiecePreview:** They handle the visualisation of the piece that appears on the screen.

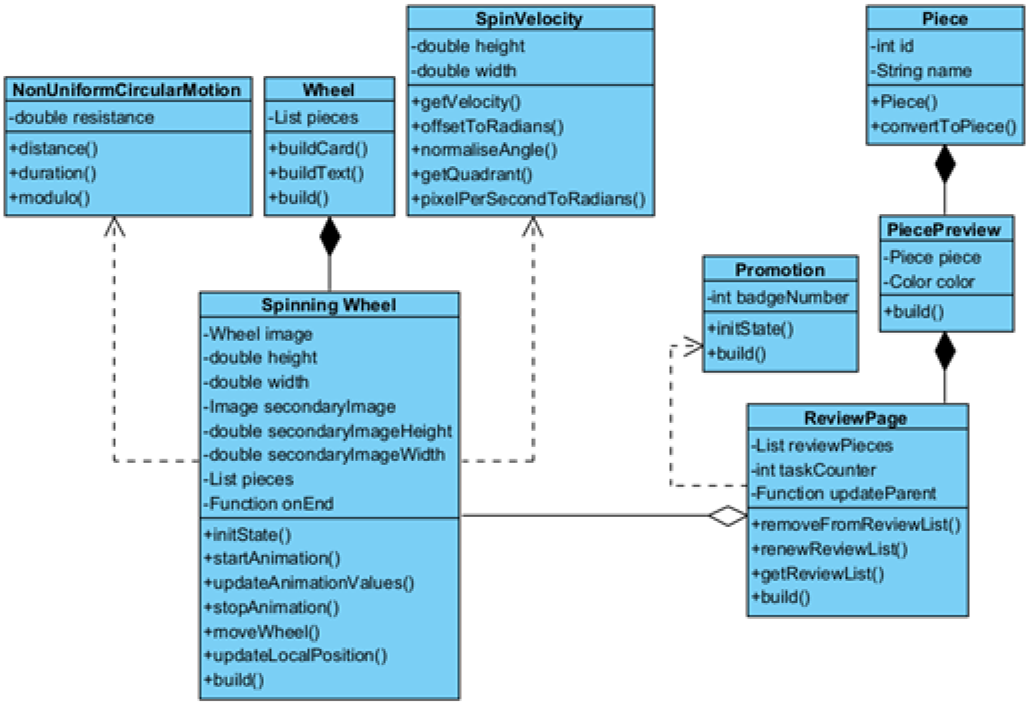


Figure 2. The class diagram of the review functionality.

Dynamic UML

Each piece has its list of review pieces. These review pieces are chosen beforehand according to the Suzuki Method and stored in the database. When the user selects the current piece, the corresponding list of review pieces is read from the database. As shown in the figure below, the list of review pieces is sent to the Wheel class. This class converts the list into the wheel displayed which is then animated.

This aspect was especially tricky to implement because of the visual changes that took place every time a new piece was selected. Since some pieces had relatively long titles that would sometimes overflow beyond the borders of the wheel, we had to set the font size of the text to a percentage of the text length.

Given the application’s target audience, it is important that the wheel looks colourful. If we randomised the choice of colours for the segments, there would be a chance of the same colours appearing twice in a row, which would be visually unappealing. To address this, we fixed the colours for each piece in advance such that this would never happen.

Making the review page was initially assumed to be a very straightforward process. We imagined that we could accomplish this using just one class. However, once we thought more about the dynamic nature, we quickly realised that this is not efficient. Hence, we adopted the structure shown below. Dividing the entire page into smaller, simpler chunks made it feasible to implement and easier to refine.

We concluded that separating the backend logic (the retrieval of the data) from the frontend visual elements (the wheel and its animations) resulted in cleaner code.

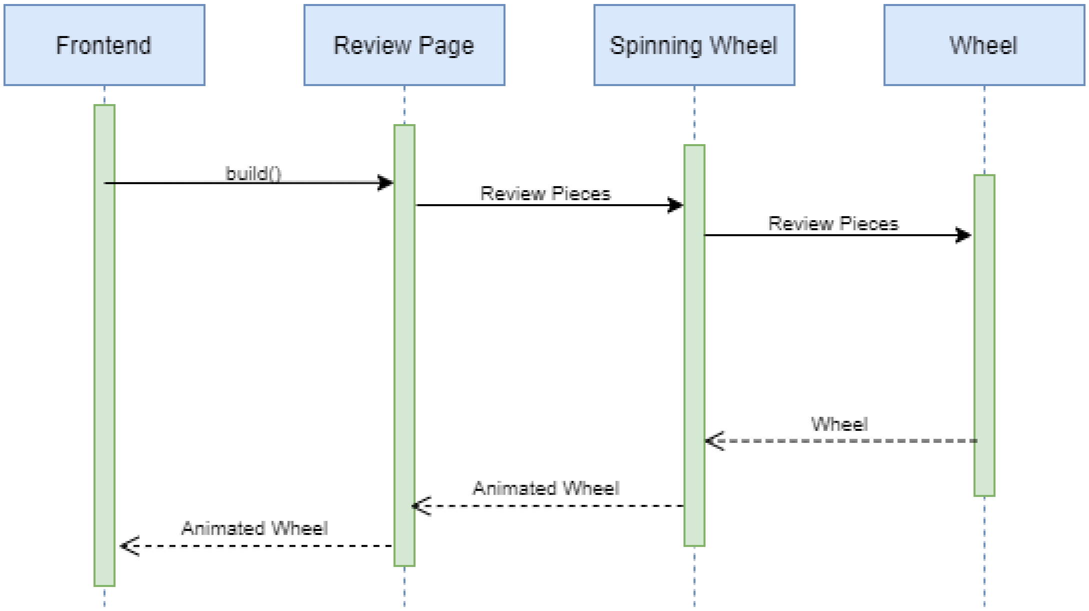
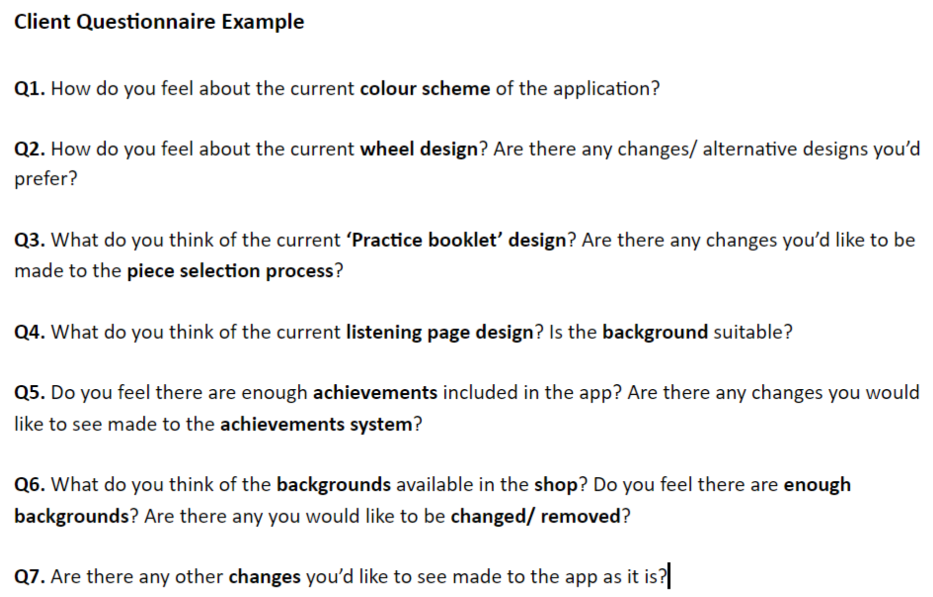


Figure 2. The sequence diagram of the review functionality.

Acceptance Testing

We frequently met with the client to review the application's progress. During these meetings, the client was given access to the latest version of the app via the Google Play Store. This enabled us to design a questionnaire and record the client's responses along with any other feedback in a document for later analysis. Questions were asked directly to the client during the call, as we felt this encouraged further discussion, as opposed to asking the client to fill in the questionnaire themselves. We chose this method of validating our product because it allowed us to regularly gather feedback from the client in a flexible, but thorough way. The questionnaire could also be tailored for each meeting, ensuring all questions were relevant and all components had the opportunity to be tested. An example questionnaire can be seen below.



Client Questionnaire Example

Q1. How do you feel about the current **colour scheme** of the application?

Q2. How do you feel about the current **wheel design**? Are there any changes/ alternative designs you'd prefer?

Q3. What do you think of the current **'Practice booklet' design**? Are there any changes you'd like to be made to the **piece selection process**?

Q4. What do you think of the current **listening page design**? Is the **background** suitable?

Q5. Do you feel there are enough **achievements** included in the app? Are there any changes you would like to see made to the **achievements system**?

Q6. What do you think of the **backgrounds** available in the **shop**? Do you feel there are **enough backgrounds**? Are there any you would like to be **changed/ removed**?

Q7. Are there any other **changes** you'd like to see made to the app as it is?

Since the client is a piano teacher, we decided that her feedback would be an adequate reflection of what teachers would think of this application. We asked for feedback from one member of each of the remaining interactor groups - parents and students. The process followed was similar, except that it was conducted in an informal setting. The participant from the "parents" group was surveyed using the same questionnaire. The younger participant playing the role of the student, on the other hand, had relatively less to say when we asked questions, so we resorted to observing how they interacted with the application.

Evaluation

Once we collected feedback from the three parties (the client, parent, and student), we held a team meeting and discussed our plans for the next release. As shown in the examples below, this feedback greatly influenced the development process, initially causing us to rethink the foundation of the entire application.

Client

Feedback received: At the very start of the development, we had misunderstood the purpose of the application. Our idea for the application was too complicated for a child to use and hence did not align with the client's request for a "fun, simple app".

Our solution: We discarded most of the application and started from scratch. Fortunately, this was during the first few weeks so we had not lost much time.

Feedback received: The number of pieces in the application is not enough.

Our solution: We addressed this by adding another book of pieces.

Feedback received: There is no way to check the progress of the student. This is important because the teacher needs to check if the student is practising regularly.

Our solution: The application awards students with a badge (starting from bronze, progressing to silver, and then gold) when they complete a task. Their final badge for any day in the current week is displayed in the progress viewer.

Student

Feedback received: We assumed that adding colours would be sufficient to make the application enjoyable to use. However, once we showed the participant our MVP, we were told that it was boring and lacked qualities that other "fun" applications have.

Our solution: After brainstorming for ideas by looking at popular applications, we settled on adding the coin system, along with the shop. Items in the shop refine the visual element by making the backgrounds colourful while giving the student an incentive to use the application more.

Parent

Feedback received: As we got closer to our final release, we decided to survey the test participants once more. Due to the plethora of features we added between releases, the participant was not sure about how to use the app or what these features do.

Our solution: We resorted to the most obvious solution - a tutorial that shows all of the features of the application. We made sure that this is displayed when the user opens the application for the first time as this would make everything clear right from the start. The tutorial can also be accessed in the settings page, which is useful if the user wants to revisit it.

Reflection

Challenges

Throughout our project, we faced a few challenges that we had to get past, both in our organisation of the project and in the development of the app. These challenges included:

- **Challenge:** Initially, we deviated a bit from the client's specification so ended up spending time developing unnecessary features that we thought were required from our interpretation of the initial brief.

  Solution: Since we discovered that we had done this, we stayed in close and frequent contact with our client to ensure she was happy with the progress we were making and the direction we were going in with the app.

- **Challenge:** Our app does not have a large number of different pages and only has a couple of main features to work on, so we struggled to distribute work evenly throughout the team without getting in each other's way with the development.

  Solution: We found that it was best if we had only some members of the team working on the development of the app, whilst others worked on project management, documentation or design elements of the app.

Processes

At the very start of the project, we decided to keep the interface simple to make it easier to implement the functionality. As we added more features to the application, we realised that the simple interface was no longer appropriate. We had to change the entire layout of the application as a result, which turned out to be a very time-consuming task. In hindsight, we should have designed the interface after considering the features we were planning to add in the future. This would have given us more time to reflect on the final structure of the application.

Impact

Our app will make the students' practise better and more productive, in turn making the teachers' jobs easier as they will not need to come up with novel ways to convince their students to practise between lessons. If the students practise better, they will improve faster, which will result in the Bristol Music School becoming known for fast development of the ability of young pianists.

The app acts as a checklist for students learning with the Suzuki Method, with a few fun features to make practising more exciting, rather than a trainer for them. Due to this, it cannot be used independently for lessons with Suzuki Method teachers, so it does not take any jobs, it only helps to improve the quality and increase the fun in lessons.