**Guitar Learning Web Application Planning**

**Problem**

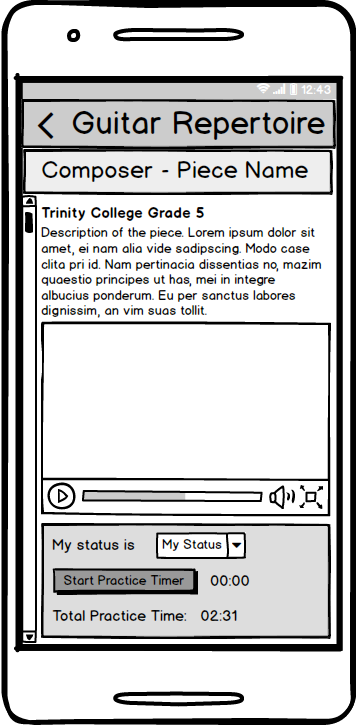
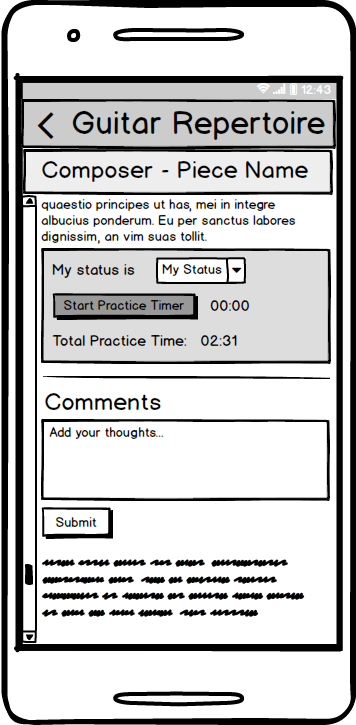
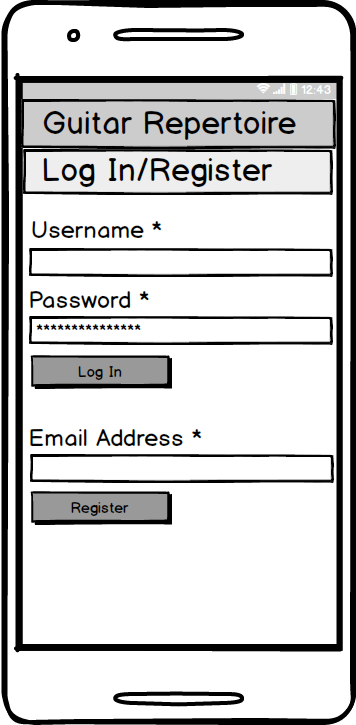
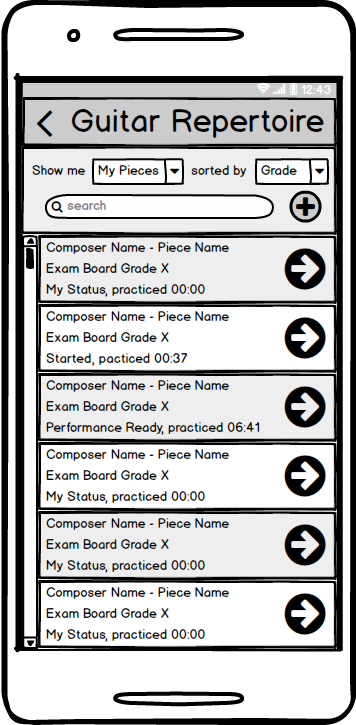
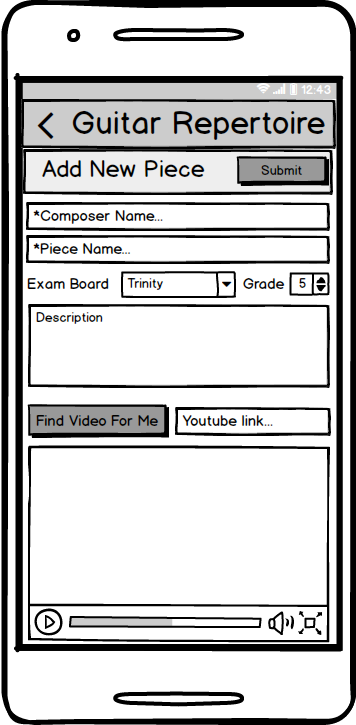
I play classical guitar and as I learn more pieces I often forget about pieces I’ve already learnt, I would like a way to keep track of my progress. I also would like a way to search for new pieces to learn of my current grade standard and be able to see what pieces are involved in each grade. I currently do this via a mix of YouTube and Spotify playlists I have made.

**Proposed Solution**

A web application that users can enter the details of a piece, difficulty level, which (if any) exam it is used in, and the embedded YouTube video. This database would be searchable by all of the above criteria. Each user would have a profile where they can tag pieces they are interested in, chart their progress on them and publicly comment on them to offer warnings and advice.

Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Name** | **Description** |
| **FR1** | Register User | The app shall be able to register an account for a user, details required are a unique username, password and email address. |
| **FR2** | Create Piece | The app shall be able to create new database entries for pieces. Fields include piece title, composer, exam board and grade (if any) from a dropdown menu, description, and a link to the piece on YouTube. Piece title and composer are required, the rest are optional. |
| **FR2.1** | Auto-populate video | The create piece page shall have a button next to the YouTube link box that sends a search query to YouTube with the piece name and composer name and returns the link for the top video result to be auto-populated in the field. The embedded video will display below the link for the user to check. |
| **FR3** | Search, Filter, Sort and Display Pieces | The pieces shall be searchable and the results sorted by name, composer, grade, exam board. |
| **FR3.1** | Search | The app will display a view that allows the user to search for pieces by text for the composer or name of the piece. |
| **FR3.2** | Filter | The search results shall have a filter option by grade, exam board, and “Piece Status” (see FR4.1). |
| **FR3.3** | Sort | The search results shall be sortable by name, composer, grade, and number of users learning. This shall be selectable as ascending and descending. |
| **FR3.4** | Display | Selecting a piece will bring up its details in a new view. |
| **FR4** | Track Repertoire | The app shall be able to connect pieces to a user account along with their status. |
| **FR4.1** | Set Piece Status | The view for a piece will include a way to select the piece status for the user. Statuses are “None”, “Interested”, “Started”, “Playable”, and “Performance Ready”. Default to “None” and save changes to user status when selected. |
| **FR4.2** | Comment | A comment box with comments by previous users shall be at the bottom on the piece page. On typing a comment and pressing the submit button the comment will be sent to the server, sanitised and displayed. |
| **FR5** | Track Practice | The app shall be able to track practice time on each piece. There will be a start/stop button on the piece page view with a timer next to it. On stopping the timer a popup will be shown asking if the user wishes to add the time spent to the total. The total time shall be displayed underneath the button. |



Scroll

**Planning and To Do**

Grey text = done

**1st step - Create a test application:**

* Setup a .Net MVC project and create models, views and controllers for “Piece” and “Exam”.
* Create a test application that uses Entity Framework code-first with a test database.
* Establish the many-to-many relationship of Exam and Piece models.
* Clean up the views:
* display the information in standard lists
* display information taken from the many-to-many relationship
* create links from the Exam Details to the Piece Details listed
* create links from the Piece Details to the Exam Details listed
* add way to update which Piece is associated with each Exam from the Piece Edit View

**References**

appdev.expert. (2018). *band music setlist.* Available: https://play.google.com/store/apps/details?id=com.noizplay.setlist. Last accessed 24th April 2018.

Cordova Plugin Vibration. (2018). *cordova-plugin-vibration.* Available: https://www.npmjs.com/package/cordova-plugin-vibration. Last accessed 24th April 2018.

Märtin, M. (2018). *Musician´s Diary.* Available: https://play.google.com/store/apps/details?id=com.vaskjala.vesiroosi20.pillipaevik. Last accessed 24th April 2018.

Payne, C. (2016). *Music Practice Pro.* Available: https://play.google.com/store/apps/details?id=www.PayneGames.MusicPracticePro. Last accessed 24th April 2018.

Simply Creative Solutions. (2017). *Practice Deck.* Available: https://play.google.com/store/apps/details?id=com.scs.practicedeck. Last accessed 24th April 2018.