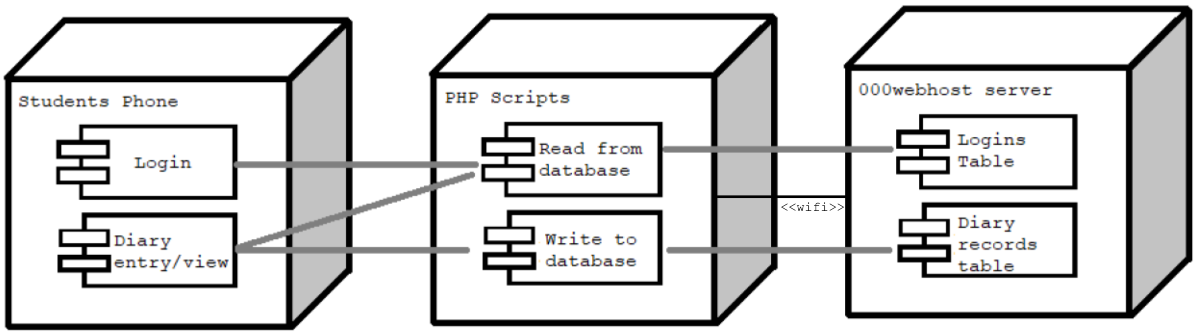
Jonathan Alderson

Software Engineering Principles

Coursework 2

Overview of Technical Design



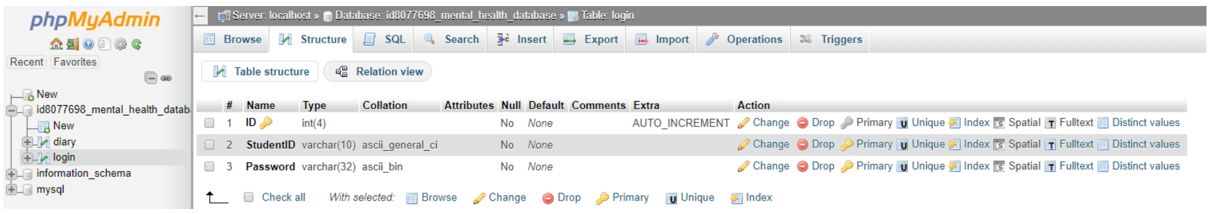
Here is the implementation diagram, for this proof. It is different from the first coursework as not all the features have been implemented yet. This current iteration uses an app on the student's phone to login and view/add diary entries which are sorted by date and stored in a remote hosted server, along with the usernames and passwords to login.

The testing for this project will be done very easily. Since my application will compile onto a phone. I can simply compile it to multiple phones and check that it works on all the phones I compile to. I have created the application with scaling in mind, so the app will work on many different aspect ratios. I can also test on different wi-fi networks, testing that data has been stored by uploading from a mobile and checking the database records from a computer. I can upload a record on account and one phone, then use another phone to see if the record has been updated or not.

From my experience from my two summer internships I have chosen to use Unity 5. It uses C# 6 for scripting and is very good at quickly making phone apps. It compiles to an APK which can be put onto a phone. Initially, I used XAMPP 3.2.2 for localhost before getting a server, using 000webhost with 10.2.12-MariaDB - MariaDB for the online server. I am also using PHP 7 for interfacing. SQL for table queries and phpMyAdmin for managing the database.

Evidence of Development

I set up my database using phpMyAdmin, first through local host, then through 000webhost. I used good practices by only creating necessary fields in tables and by making reasonable sized inputs (such as name having a maximum length of 10) to avoid waste. I also included automatically incrementing ID numbers for both tables to make it easy to identify records for the diary.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(4)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	StudentID	varchar(10)	ascii_general_ci		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	Password	varchar(32)	ascii_bin		No	None			Change Drop Primary Unique Index Spatial Fulltext Distinct values

For the PHP scripts that interface with the server I programmed them in Atom. There are two scripts, one which reads all the information from the server. And one which allows data to be inputted to the diary table in the database.

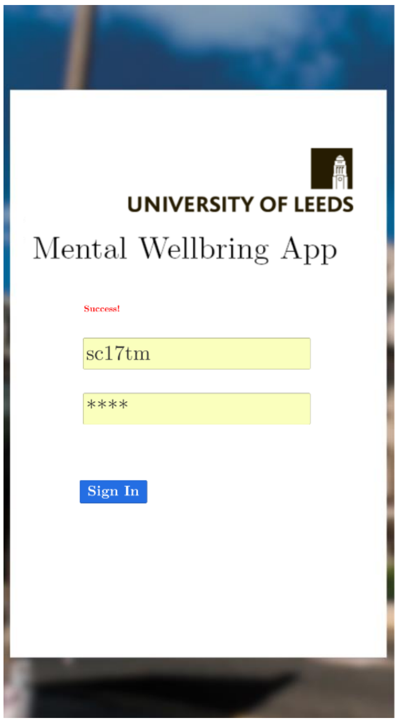
This is part of the code from the reading section, from these initial variables, I have used good practices by using a long complex password, the database and username have relevant names. (The id8077698 bit is needed for the server hosting I am using). The rest of the script calls the SQL commands shown below and echo's them to the screen to be used later by Unity.

```
<?php

  // Database Variables
  $servername = "localhost";
  $username = "id8077698_jonathanalderson";
  $password = "mqsvPGizz4MJJgN";
  $dbName = "id8077698_mental_health_database";

  . . .
  // These commands get all the data from the two tables
  $sql = "SELECT * FROM login";
  $sql2 = "SELECT * FROM diary";
// These commands echo each entry in the table
echo "ID:".$row['ID']."|StudentID:".$row['StudentID']."|Password:".$row['Password']
. ";";
echo "ID:".$row['ID']."|StudentID:".$row['StudentID']."|Date:".$row['Date']."|Log:"
.$row['Log']."|LogTitle:".$row['logTitle']. ";";
```

For the main program. I used Unity, I made background images for all the screens using GIMP, I then used the Unity canvas to organise all the screens and the UI elements for them. I then created 9 scripts to run each of the buttons that I needed. I used good design principles by using a similar login to Minerva, so that it was familiar. I also have confirmation text in multiple places in the app, to the user can see if their login/diary log has been successful, along with error messages to fix it. In my code I also made use of clear public variables setup when the initial database is read that can be used by other scripts, to make the system easy to expand and modular. For testing the app, I compiled it onto my phone, added new records and checked that it updated the database from my computer. I then relaunched the app to see that the new data was still there. Also, I ran the app on two different phones, both logged in on the same account, updating a log on phone 1, then refreshing phone 2 to see that the information had changed.

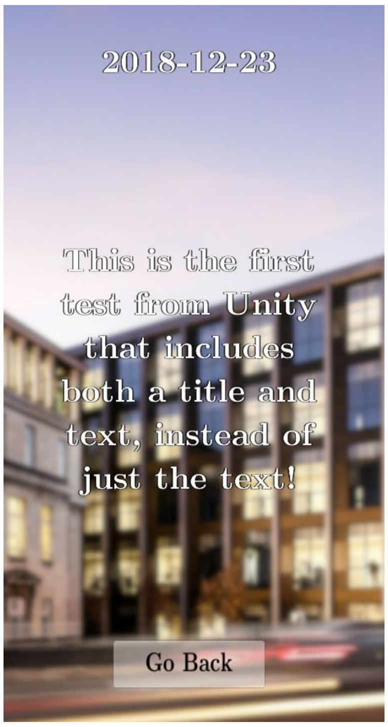

UNIVERSITY OF LEEDS

Mental Wellbring App

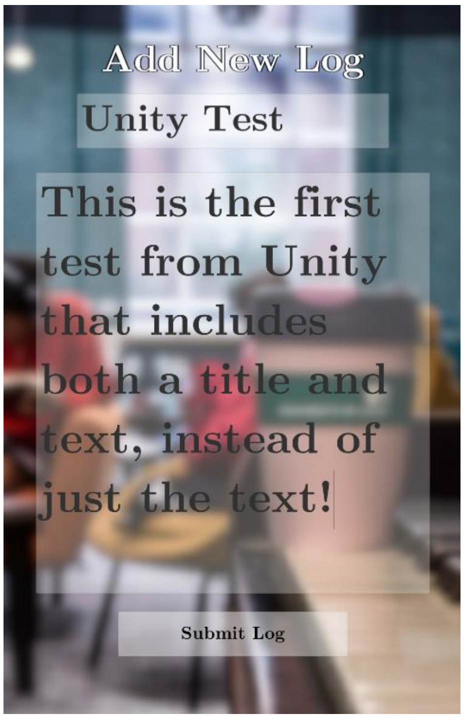
Success!

sc17tm

****

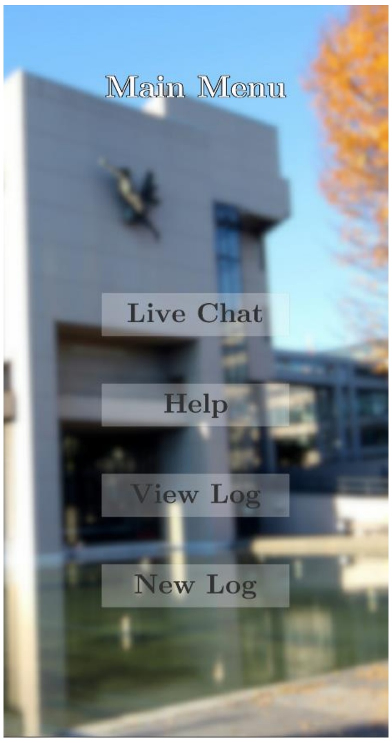

2018-12-23

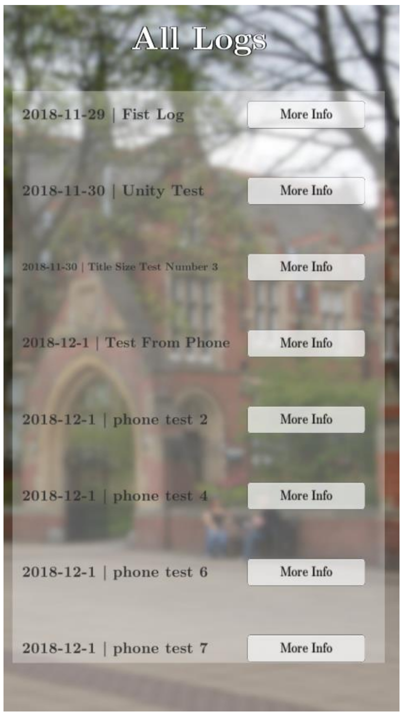
This is the first test from Unity that includes both a title and text, instead of just the text!


Add New Log

Unity Test

This is the first test from Unity that includes both a title and text, instead of just the text!




All Logs

2018-11-29 | Fist Log — More Info

2018-11-30 | Unity Test — More Info

2018-11-30 | Title Size Test Number 3 — More Info

2018-12-1 | Test From Phone — More Info

2018-12-1 | phone test 2 — More Info

2018-12-1 | phone test 4 — More Info

2018-12-1 | phone test 6 — More Info

2018-12-1 | phone test 7 — More Info

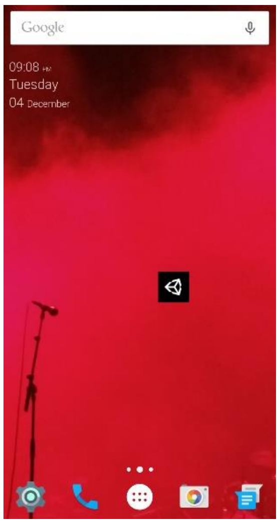
The ‘all logs’ section makes use of a scrolling list. This is good because it allows as many logs to fit on the phone. The date and title of the log is shown for to make the app more usable. Because of the way the app is made, it would be very easy to change the background images and could be customised by each user. This is an addition that could be done in the future.

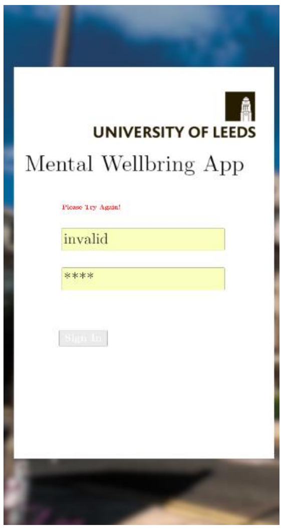
```
1.  public class updateLog : MonoBehaviour {
2.
3.      // public inports
4.      public DataLoader data;
5.      public loginInformation loginInfo;
6.      public GameObject TemplateLog;
7.      public Transform parent;
8.      private List<DiaryEntry> allLogs;
9.
10.     void OnEnable()
11.     {
12.         // ran every time the page changes
13.         allLogs = data.diaryEntries;
14.         int j = parent.childCount;
15.         int m = 0;
16.         int n = 0;
17.         // destroys every diary input
18.         for(int k = 1; k < j; k++)
19.         {
20.             Destroy(parent.GetChild(k).gameObject);
21.         }
22.         // creates a new log with the arguments from the datastruct
23.         for (int i = 0; i < allLogs.Count; i++)
24.         {
25.             if(allLogs[i].UserID == loginInfo.currentLogin)
26.             {
27.                 NewLog(allLogs[i].Year + "-" + allLogs[i].Month + "-
    " + allLogs[i].Day + " | "  + allLogs[i].LogTitle, allLogs[i].ID);
28.             }
29.         }
30.     }
31.
32.     public void NewLog(string text, int id)
33.     {
34.         GameObject NewLog = Instantiate(TemplateLog, parent);
35.         // gives each button an ID which can be passed around
36.         NewLog.GetComponent<logID>().logIDNumber = id;
37.         NewLog.transform.GetChild(0).GetComponent<Text>().text = text;
38.         NewLog.GetComponent<RectTransform>().anchorMin -= new Vector2(0, 0.2f);
39.         NewLog.GetComponent<RectTransform>().anchorMax -= new Vector2(0, 0.2f);
40.         NewLog.GetComponent<RectTransform>().rect.Set(0, 0, 0, 0);
41.     }
42. }
```

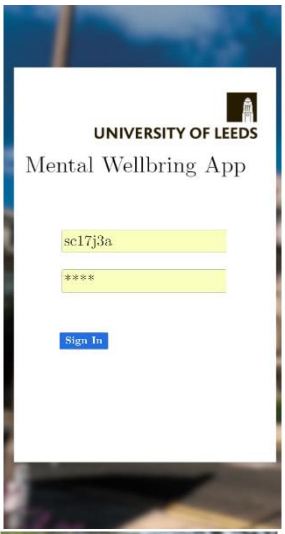
Here is some example code from Unity. This is the script that controls the scrollable list. It is dynamic so can change in size. The function is only ran every time the page turns on, as not to waste resources. The global on line 4 is the data structure of all the current diaries. This is handled by another script and is only updated when the user adds a new record. This is efficient as the server does not always have to be called to. A function has been made to create each item in the list to increase readability, lines 36-40 handle the position of the item, as it needs to be adjusted. Once a new item is made, the title it shows and the link that the button goes to is updated to reflect the data it links us to. The constructor is nicely made so for each new item in the list, we can just look at the information from the allLogs[] list that we have access to.

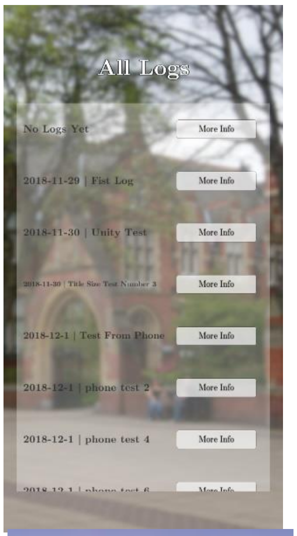
Evidence of deployment

https://goo.gl/abz7Ap Linked is a video recorded on my phone running through the main functionality. This is not being ran off a computer and none of the log information is being saved on the phone. The server had been running for a few days as of recording the video.

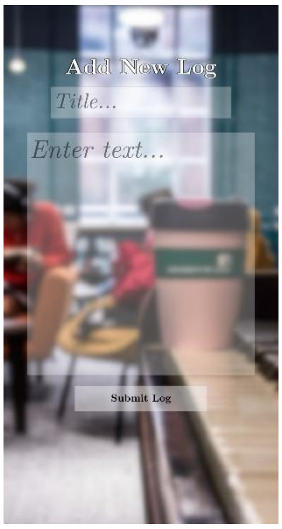


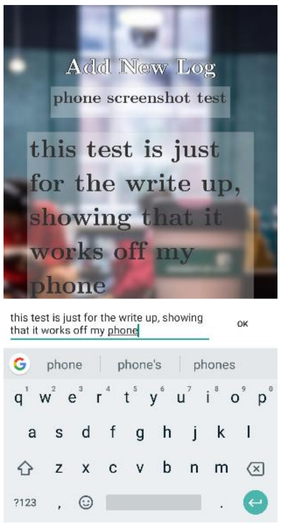

UNIVERSITY OF LEEDS
Mental Wellbring App
Please Try Again!
invalid
****
Sign In


UNIVERSITY OF LEEDS
Mental Wellbring App
sc17j3a
****
Sign In

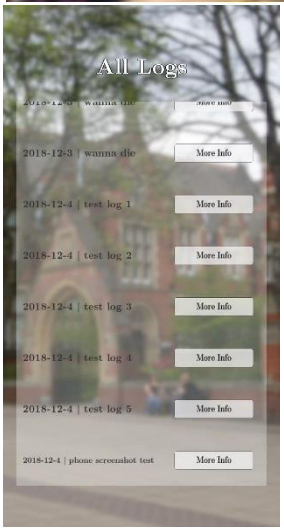

All Logs

No Logs Yet | More Info
2018-11-29 | Fist Log | More Info
2018-11-30 | Unity Test | More Info
2018-11-30 | Title Size Test Number 3 | More Info
2018-12-1 | Test From Phone | More Info
2018-12-1 | phone test 2 | More Info
2018-12-1 | phone test 4 | More Info
2018-12-1 | phone test 6 | More Info

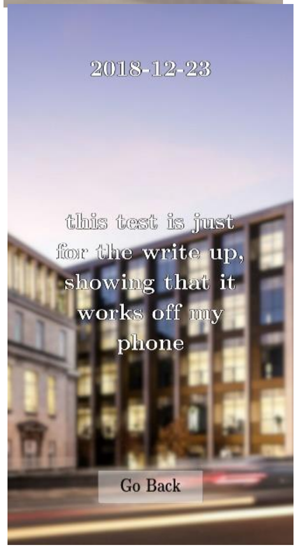

Add New Log
Title...
Enter text...
Submit Log


Add New Log
phone screenshot test
this test is just for the write up, showing that it works off my phone

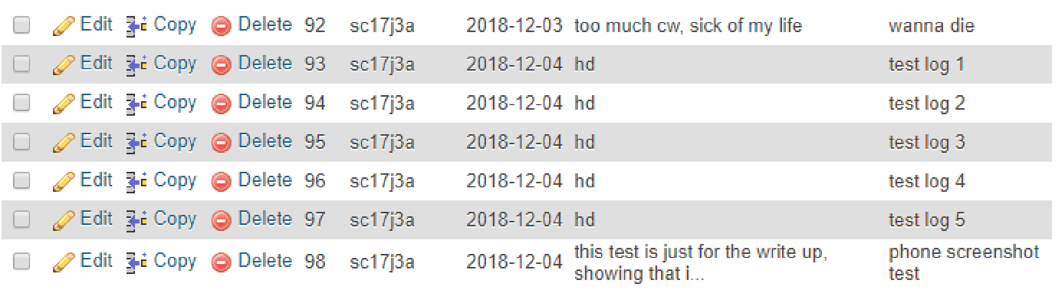
this test is just for the write up, showing that it works off my phone
OK


All Logs

2018-12-3 | wanna die | More Info
2018-12-3 | wanna die | More Info
2018-12-4 | test log 1 | More Info
2018-12-4 | test log 2 | More Info
2018-12-4 | test log 3 | More Info
2018-12-4 | test log 4 | More Info
2018-12-4 | test log 5 | More Info
2018-12-4 | phone screenshot test | More Info


2018-12-23
this test is just for the write up, showing that it works off my phone
Go Back

These eight pictures are all screenshots from my phone. There were done to test the basic functionality of the app. It is a similar routine to the video, but now in screenshot form. The first screenshot shows the app installed on my phone, it was able to compile with no errors and ran well after being installed. The second screenshot shows me entering an erroneous login, an error message is given to me and I am asked to try again. The third screenshot shows me entering a valid login. When I press sign in, I am taken to the main menu. Because we have been taken to the main menu. I know that the database reading has worked. Since none of the user logins are stored on the phone, I know that the database has been successfully read from. The server has been up for a week or so now, so this is good news. Upon previous attempts the login would not work. This however, is because my database is quite slow sometimes, I put this down to the server being free, as sometimes read/write times would be more than 5 minutes, and others be 0.01 seconds. If this app was implemented, it would be using the university servers, so this would not be an issue. The main menu has not been shown in these screenshots but can be seen on the page above. On the phone, all the buttons work, help and live chat do not lead to anything as they have not been implemented yet. Screenshot 4 shows all the logs for the current user, these have all been downloaded from the server successfully, all the logs are for the correct user, and no logs from any other user can be seen. Screen shot 5 shows the screen for entering a new log. Both input boxes worked correctly on mobile and brought up the correct keyboard. When entering an error message is given if the boxes are left blank, and a success message is given when everything is fine, and the record has been written to the database. Screenshot 8 shows the updated version of the logs since now a new record has been added. The final screenshot below shows the view of the database from the browser. And we can see that the new log has been added and displayed on there.



☐	Edit	Copy	Delete	92	sc17j3a	2018-12-03	too much cw, sick of my life	wanna die
☐	Edit	Copy	Delete	93	sc17j3a	2018-12-04	hd	test log 1
☐	Edit	Copy	Delete	94	sc17j3a	2018-12-04	hd	test log 2
☐	Edit	Copy	Delete	95	sc17j3a	2018-12-04	hd	test log 3
☐	Edit	Copy	Delete	96	sc17j3a	2018-12-04	hd	test log 4
☐	Edit	Copy	Delete	97	sc17j3a	2018-12-04	hd	test log 5
☐	Edit	Copy	Delete	98	sc17j3a	2018-12-04	this test is just for the write up, showing that i...	phone screenshot test

Evaluation of design

From my tests, my proof of architecture has done very well. It can compile onto android phones and works on all standard phone screen aspect ratios. The app runs efficiently, even on older phones (tested on Moto G2 with no issues). The list containing all your logs does not slow down, even when displaying a lot of items and the app is very usable. The UI design is clean, using stylish backgrounds and semi-transparent elements, I negated the use of a back button on most screens (that would take up space) by mapping the phones built in back button to take you back to the menu.

There were challenges with Unity, thinking about a clean way to program things with all the public variables I would have. I ended up creating a local duplicate of all the logs from the database contained within a Diary data structure that I made. This contains all the things you need to display and maintain a record. This means that the server does not need to be contacted as often, and when it is, the diary information is updated. It was also a challenge transferring my database from a localhost using XAMPP to an actual online one using 000webhost. I ended up looking around until I found a free one and followed a tutorial to transfer my tables and PHP files, this was difficult as I had not done any real database implementation before. To improve the app and improve security, I would put each students records in a separate table, this would improve read speeds once the table got sufficiently large and in each table I would encrypt the data and have the decryption key tied to a phone on first time setup, so that it could only be read from one place.