



University of Ottawa

Faculty of Engineering

**GNG2101 Introduction to product Development and management
for Engineers**

Project Deliverable F

Submitted by:

Talk Box. C01, Team C13

Tia El Masry, 300160596

Zainab Badawi, 300034146

Kain Mozafarian, 300138481

March 7, 2021

University of Ottawa

Table of Contents

Table of Contents	i
List of Figures	iii
List of Tables	iv
List of Acronyms	Error! Bookmark not defined.
1 Introduction.....	5
2 Client Feedback	6
2.1 Client Feedback Keynotes:.....	6
2.2 Feedback Analysis:	6
3 Prototype 2 Diagrams/Pictures	7
3.1 Software:	9
3.2 Hardware:.....	12
3.2.1 Initialization	12
3.2.2 MYSQL Database.....	12
3.2.3 PHPMYAdmin.....	12
3.2.4 LCD Screen.....	14
3.2.5 Button and Speaker	18
3.2.6 joystick.....	20
4 Prototype Testing & Evaluation.....	23
5 Conclusions and Recommendations for Future Work	26
6 Bibliography	Error! Bookmark not defined.
APPENDICES	27

APPENDIX I:	27
References.....	27

List of Figures

Figure 1: Talk Box Functionality Breakdown	8
Figure 2: Python script that creates the GUI using Tkinter	11
Figure 3: PhpMyAdmin to login page	13
Figure 4: Editing the database.....	13
Figure 5: User main display panel	15
Figure 6: “Home Devices” phrases displayed as buttons, ready to be voiced via speaker.....	16
Figure 7: “Simple Phrases” word bank displayed as buttons, ready to be voiced via speaker	17
Figure 8: pushbutton.py script	18
Figure 9: Button SolidWorks Design.....	19
Figure 10: 3D printed and Button mounted	19
Figure 11: Full assembly of the button	20
Figure 12: Joystick extension SolidWorks cross section	21
Figure 13: 3D printed joystick extension.....	22
Figure 14: Hardware prototype parts	23

List of Tables

Table 1: Target Specifications Comparison..... 24

1 Introduction

In this deliverable, group C13's goals are to demonstrate their understanding and showcase their knowledge by creating a second iteration of prototypes, with regards to the client feedback. At this stage of the project, the group is orienting themselves around doing software prototypes for the most part, whilst refining older prototypes such as readjusting the shape of the input method, using a different screen size display, and using a Raspberry Pi for our microcontroller instead of an Arduino. In this stage of the project, we are focused on prototyping and testing out methods regarding the program that will be running on the microcontroller, it will support phrases being displayed, text to speech, and images that are in accordance with their respective phrases. A method to connect the microcontroller to another device node, which is able to update data on the microcontroller wirelessly is another scope of our prototyping procedure.

2 Client Feedback

Interpret client comments and feedback into meaningful criteria to be used to make or assist in making prototype 2. Aspects that are liked or are given counterexamples are clearly extracted.

2.1 Client Feedback Keynotes:

- XLR cable to power the the arduino
- the wheelchair runs on 24 volts
- The analog stick should move left and right horizontally, one direction for navigation and the other for selection
- Use a “gooseneck” design for the support of the display
- The Google Assistant is not as necessary as the main text to speech functionality, probably shouldn’t be pursued any further
- Use a larger screen for the user display
- The software is adaptable to smaller screen sizes for displaying the python app (smaller compared to tablet sized screens)
- Limited space on a wheelchair, so a different input interface needs to be thought of
- Reduce the length of the joystick mold to reduce stress on the joystick circuit itself (about ~5 cm)
- Create a panelling system/ include tabs where phrases are shown in the most user-friendly fashion possible

2.2 Feedback Analysis:

Extracts interesting and useful insight from feedback while identifying how this information can be used to improve the solution.

The main key takeaways from the client meeting were in regard to the software and the input method of the overall system. Whilst demonstrating the physical layout of the Talkbox, the client showed concern regarding screen size, screen mounting mechanism, and space pollution

caused by multiple devices being present. A further explanation was given by us regarding some aspects of the hardware; first of, we displayed that the monitor that will be used will be different than the one demonstrated in the first prototype, that the original monitor was used in accordance to using an Arduino for our microcontroller, since now it has been decided that further prototyping will be done in regards with a Raspberry Pi. A button and a joystick turned out to be a hectic solution according to the client, so the suggested workaround is to have only the joystick which enacts as a bidirectional input response. Furthermore, the client had suggested a Gooseneck mount for our display in order to allow free movement of the display with some rigidity in mind. As for the software part of the project, we showcased our progression from in-line command interface from prototype 1, to a work in progress version in prototype 2 where the program reads a text file and showcases phrases in a button that's displayed on a GUI. The program can go backwards and forwards in terms of what category of phrases it is wanted by the user. Furthermore, the array of phrases is updated in the program without needing to terminate it after updating the text file is edited. A few concerns regarding the software from the client was how the program's GUI would fit on a small screen and how would multiple phrases fit on one panel. These are issues that we will tackle in the next few weeks.

3 Prototype 2 Diagrams/Pictures

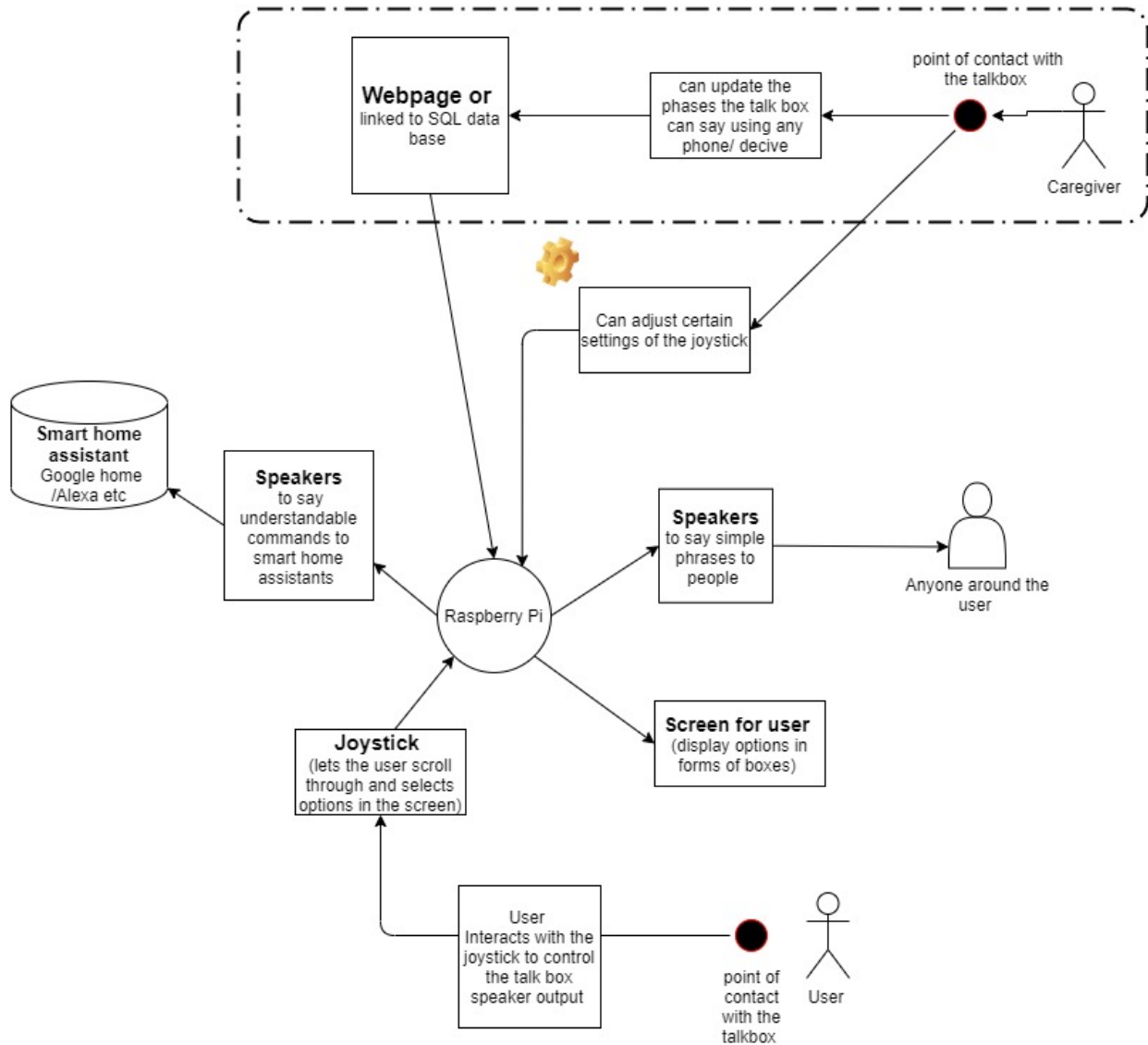
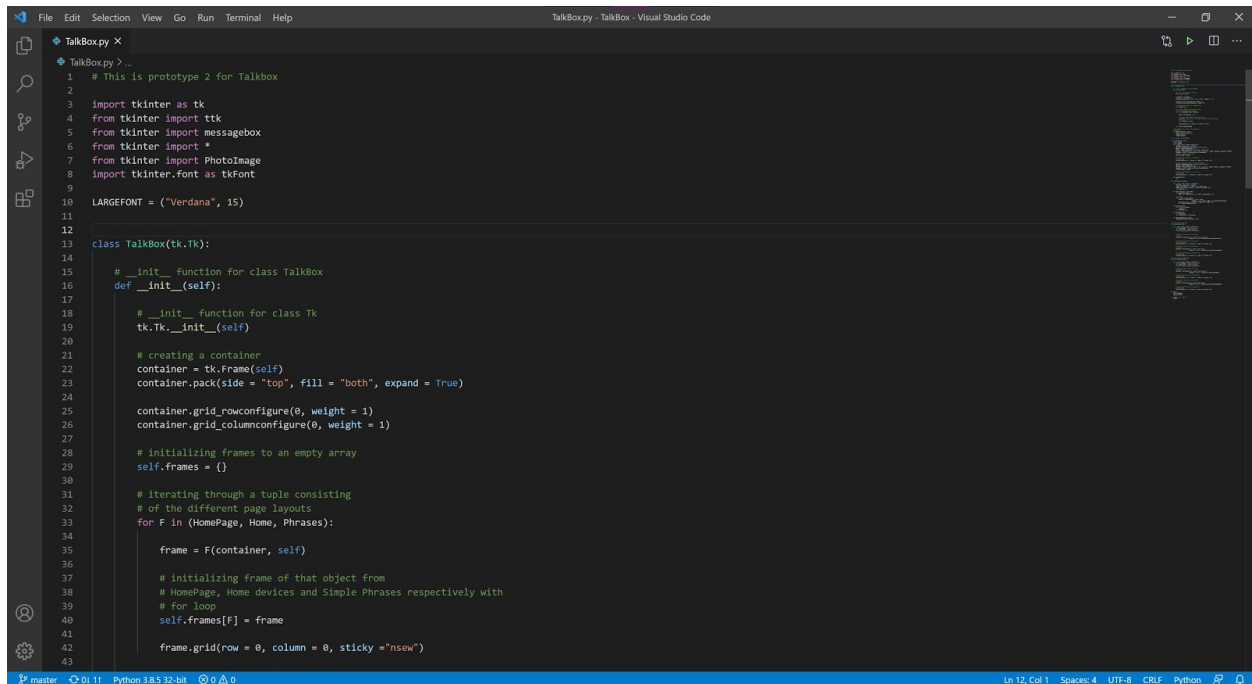


Figure 1: Talk Box Functionality Breakdown

This Prototype objective is to establish functionalities that allow the raspberry Pi to output sound, and display options for the user to pick between speaking with devices or people (shown in **Figure 1**: user main display panel). In addition is to allow the user to navigate between options by moving the joystick in the x-axis then selecting the option by pushing on the joystick downwards. Finally, caregiver is able to add phrases to the sql data phase through a webpage that is hosted in one of the group members existing webpage server. (**Figure**: Talk Box Functionality Breakdown) demonstrate a detailed breakdown of the talk box functionalities.

3.1 Software:

In terms of the software prototype, we have created a python script that creates a graphical user interface with the Tkinter package. We have created 5 different classes that interact with each other: the TalkBox class, the HomePage class, the BaseFrame class, the Home class and the Phrases class. The TalkBox class acts as the initializer for the frame of the entire program which creates the frames for every page created in the class. The HomePage class contains two widget buttons that have photo images, where their commands are to go to their respective pages: “Talk to devices” which opens the Home class and “Talk to people” which opens the Phrases class. The most important class is the BaseFrame class which acts as a parent to both the Home class and the Phrases class since it contains all the common functions between both classes. This class contains the function that reads the text files and creates widget buttons from all the entries from those files. The commands for those buttons are currently creating a message show box that returns the specific sentence selected. For the future, we are planning to replace those current commands with code to send to the text-to-speech API which will voice those sentences. In our BaseFrame, we also have a refresh function which allows the pages to get automatically updated with buttons when the text files are updated, without having to kill the program.



```
1 # This is prototype 2 for Talkbox
2
3 import tkinter as tk
4 from tkinter import ttk
5 from tkinter import messagebox
6 from tkinter import *
7 from tkinter import PhotoImage
8 import tkinter.font as tkFont
9
10 LARGEFONT = ("Verdana", 15)
11
12
13 class TalkBox(tk.Tk):
14
15     # __init__ function for class TalkBox
16     def __init__(self):
17
18         # __init__ function for class Tk
19         tk.Tk.__init__(self)
20
21         # creating a container
22         container = tk.Frame(self)
23         container.pack(side = "top", fill = "both", expand = True)
24
25         container.grid_rowconfigure(0, weight = 1)
26         container.grid_columnconfigure(0, weight = 1)
27
28         # initializing frames to an empty array
29         self.frames = {}
30
31         # iterating through a tuple consisting
32         # of the different page layouts
33         for F in (HomePage, Home, Phrases):
34
35             frame = F(container, self)
36
37             # initializing frame of that object from
38             # HomePage, Home devices and Simple Phrases respectively with
39             # for loop
40             self.frames[F] = frame
41
42             frame.grid(row = 0, column = 0, sticky = "nsew")
43
```

```

File Edit Selection View Go Run Terminal Help
TalkBox.py - TalkBox - Visual Studio Code

TalkBox.py X
TalkBox.py > ...
42         frame.grid(row = 0, column = 0, sticky = "nsew")
43
44         self.show_frame(HomePage)
45
46         # to display the current frame passed as
47         # parameter
48         def show_frame(self, cont):
49             frame = self.frames[cont]
50             frame.refresh()
51             frame.tkraise()
52
53         # first window frame HomePage
54
55         class HomePage(tk.Frame):
56             global photo1
57             global photo2
58             def __init__(self, parent, controller):
59                 tk.Frame.__init__(self, parent)
60                 photo1 = PhotoImage(file = "HomeDevicesPic.gif")
61                 photo1 = photo1.subsample(3,3)
62                 style = tkFont.Font(family = "Verdana", size = 15)
63                 button1 = ttk.Button(self, text = "Talk to devices", image = photo1, compound = BOTTOM,
64                                     command = lambda : controller.show_frame(Home))
65                 #button1.config(font = style)
66                 button1.image = photo1
67
68                 # putting the button in its place by
69                 # using grid
70                 button1.grid(row = 1, column = 1, padx = 10, pady = 10)
71
72                 ## button to show frame 2 with text layout2
73                 photo2 = PhotoImage(file = "SimplePhrasesPic.gif")
74                 photo2 = photo2.subsample(3,3)
75                 button2 = ttk.Button(self, text = "Talk to people", image = photo2, compound = BOTTOM,
76                                     command = lambda : controller.show_frame(Phrases))
77                 button2.image = photo2
78
79                 # putting the button in its place by
80                 # using grid
81                 button2.grid(row = 1, column = 5, padx = 10, pady = 10)
82
83         def refresh(self):
84             pass

```

```

File Edit Selection View Go Run Terminal Help
TalkBox.py - TalkBox - Visual Studio Code

TalkBox.py X
TalkBox.py > ...
86         class BaseFrame(tk.Frame):
87
88             def __init__(self, parent, controller):
89                 tk.Frame.__init__(self, parent)
90                 label = ttk.Label(self, text = self.label_text,)
91                 label.grid(row = 0, column = 2, padx = 10, pady = 10)
92                 self.buttons = []
93
94             def read_file(self, file_name):
95                 with open(file_name) as f:
96                     data = {i: line.strip() for i, line in enumerate(f, 1)}
97
98                 # buttons
99                 for i, name in data.items():
100                     button = ttk.Button(self, text = name,
101                                       command = lambda name = name: self.show_message(name))
102                     button.grid(row = 2, column = i, padx = 10, pady = 10)
103                     self.buttons.append(button)
104
105             def clean_up(self):
106                 for i in self.buttons:
107                     i.destroy()
108                 self.buttons = []
109
110             def refresh(self):
111                 self.clean_up()
112                 self.read_file(self.file_name)
113
114             def show_message(self, text):
115                 messagebox.showinfo(message = text)
116
117
118         # second window frame Home
119         class Home(BaseFrame):
120
121             def __init__(self, parent, controller):
122                 self.file_name = "Home Devices.txt"
123                 self.label_text = "Home Devices"
124                 super().__init__(parent, controller)
125
126                 # button to show frame 2 with text
127                 # layout2

```

```
File Edit Selection View Go Run Terminal Help TalkBox.py - TalkBox - Visual Studio Code
TalkBox.py X
TalkBox.py >
127 # button to show frame 2 with text
128 # layout2
129 button1 = ttk.Button(self, text="Simple phrases",
130                      command = lambda : controller.show_frame(Phrases))
131
132 # putting the button in its place
133 # by using grid
134 button1.grid(row = 3, column = 1, padx = 10, pady = 10)
135
136 # button to show frame 2 with text
137 # layout2
138 button2 = ttk.Button(self, text="Home page",
139                      command = lambda : controller.show_frame(HomePage))
140
141 # putting the button in its place by
142 # using grid
143 button2.grid(row = 3, column = 2, padx = 10, pady = 10)
144
145 # third window frame Phrases
146 class Phrases(BaseFrame):
147
148     def __init__(self, parent, controller):
149         self.file_name = "Simple Phrases.txt"
150         self.label_text = "Simple Phrases"
151         super().__init__(parent, controller)
152
153         # button to show frame 2 with text
154         # layout2
155         button1 = ttk.Button(self, text="Home devices",
156                             command = lambda : controller.show_frame(Home))
157
158         # putting the button in its place by
159         # using grid
160         button1.grid(row = 3, column = 1, padx = 10, pady = 10)
161
162         # button to show frame 3 with text
163         # layout3
164         button2 = ttk.Button(self, text="Home Page",
165                             command = lambda : controller.show_frame(HomePage))
166
167         # putting the button in its place by
168         # using grid
169         button2.grid(row = 3, column = 2, padx = 10, pady = 10)
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
269
```

3.2 Hardware:

3.2.1 Initialization

For the initialization process, the Raspberry Pi Imager was used to flash the SD card with the “Raspberry Pi OS (32-bit)”. Following that, the Pi was connected to a monitor using a HDMI cable and the country, language, time zone, password, and wifi were set up.

3.2.2 MYSQL Database

The team used MYSQL to set up a database on the Raspberry Pi. A tutorial made by PiMyLife was used to complete the database setup. The tutorial guided through the process of installing “mysql_secure_installation”, followed by creating the database user with the following commands: “sudo mysql -u root -p” , CREATE “DATABASE TalkBox;” , “CREATE USER zainab@localhost IDENTIFIED BY 'pimylifeup;'” and all the privileges were granted to the zainab@localhos user.

3.2.3 PHPMyAdmin

PHPMyAdmin was used to administer the MySQL database created, using a similar tutorial by PiMyLife. A summary of the steps done include: on the Raspberry terminal PHPMyAdmin “sudo apt install phpmyadmin”. Secondly, the apache was configured for PHPMyAdmin. Thirdly, NGINX was configured for PHPMyAdmin. The current prototype uses PHPMyAdmin database that is hosted on the local server. However, the team is working on hosting it on a GoDaddy server to allow users and carveger to edit the database from any device on any network.

The figure “PhpMyAdmin to login page” shows the initial page to access the database.

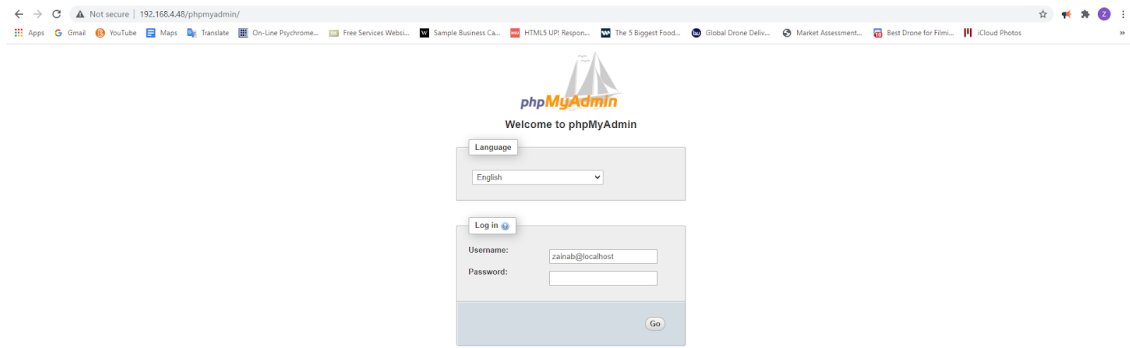


Figure 3: PhpMyAdmin to login page

Two tables were created, “HomeDevices” and “SimplePhrases” and were populated with phrases that are directed to either home assistant devices or people.

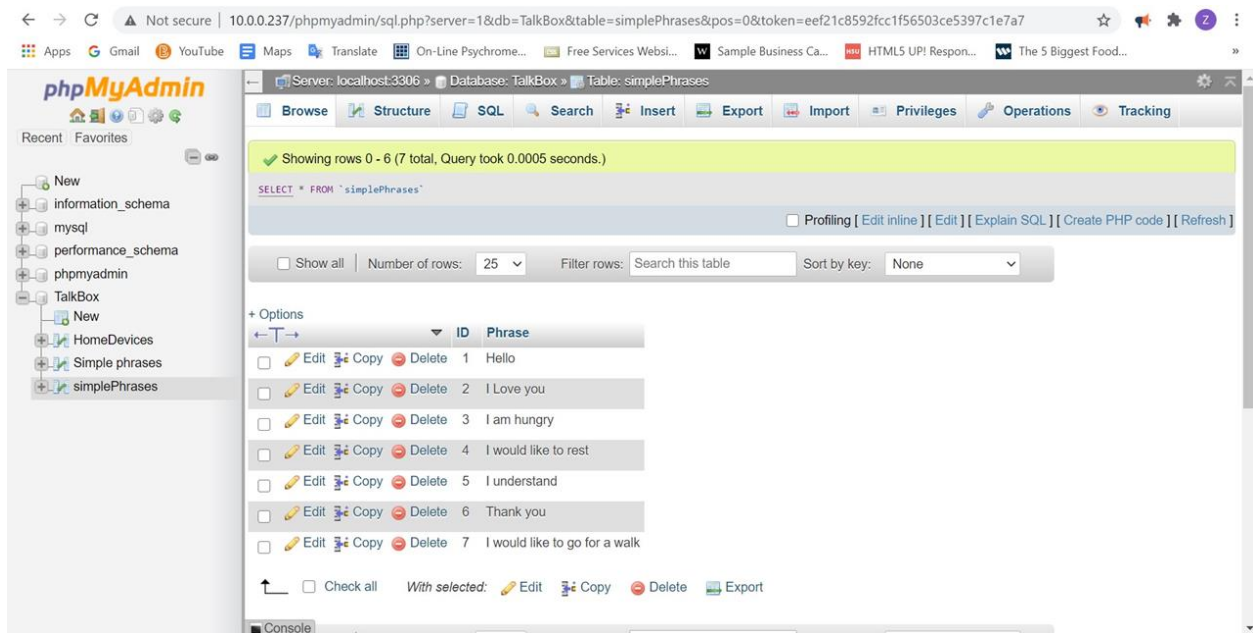


Figure 4: Editing the database

3.2.4 LCD Screen

A 3.5-inch LCD screen is used to display the phrase options to the user. The following five lines were typed in the terminal to set up the LCD screen:

- `sudo rm -rf LCD-show`
- `git clone https://github.com/goodtft/LCD-show.git`
- `chmod -R 755 LCD-show`
- `cd LCD-show/`
- `sudo ./LCD35-show`

The TalkBox github repository was cloned to the Raspberry Pi, and the talkbox.py python script was successfully executed. The talkbox.py script outputs the following panel, where the user can first choose if they want to output phrases that talk to people or talk to devices as shown in the **figure 5** “User main display panel”. After the user selects one of the options, they will be directed to another panel that provides multiple buttons with phrases in the database, as shown in the **figure 6** “Home Devices” phrases displayed as buttons, ready to be voiced via speaker’, and **figure 7** “Simple Phrases” word bank displayed as buttons’, ready to be voiced via speaker. Currently the phrases shown are stored in a text file in the TalkBox github repository. But in the following weeks, the team will create a query that fetches data directly from the phpMyAdmin hosted on GoDaddy server.

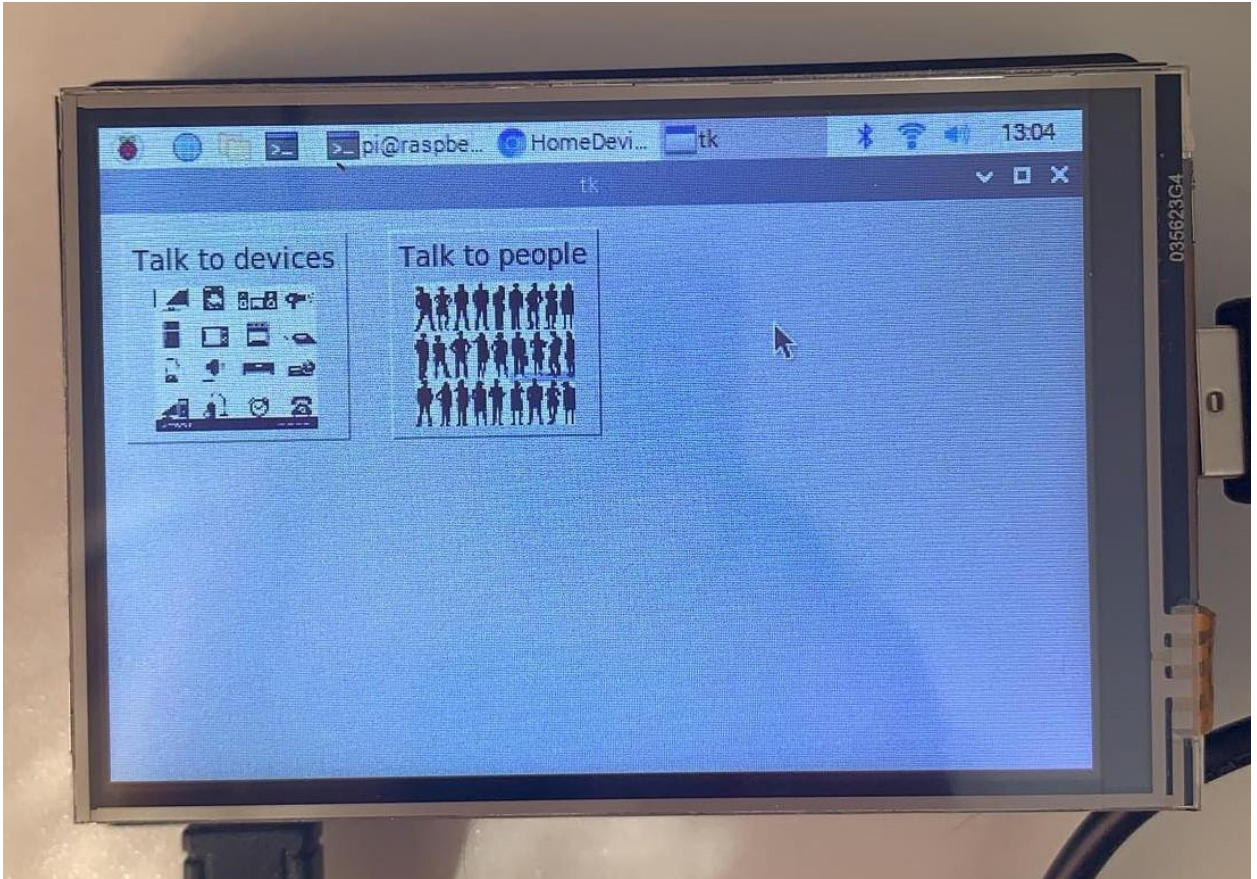


Figure 5: User main display panel



Figure 6: “Home Devices” phrases displayed as buttons, ready to be voiced via speaker

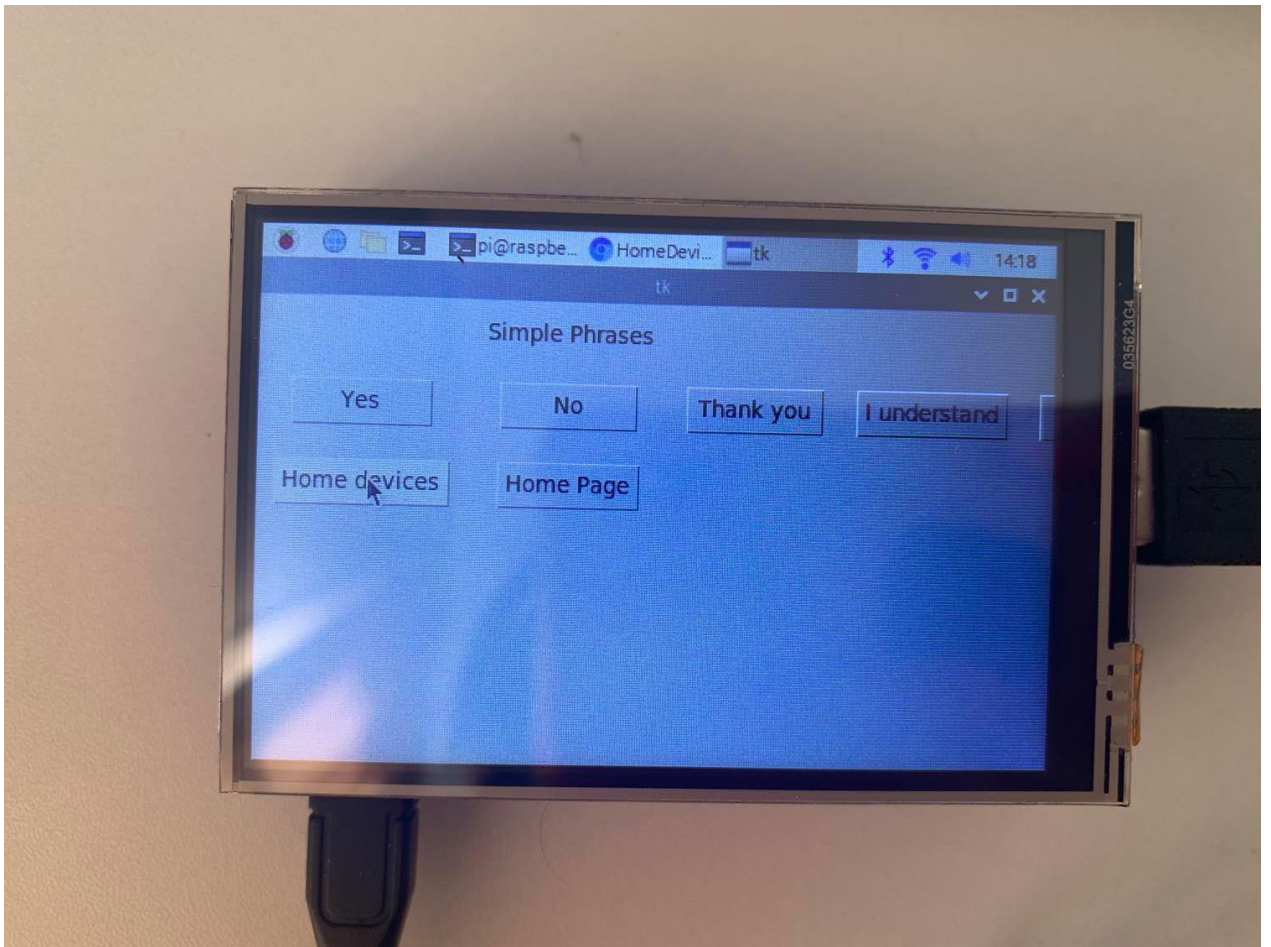
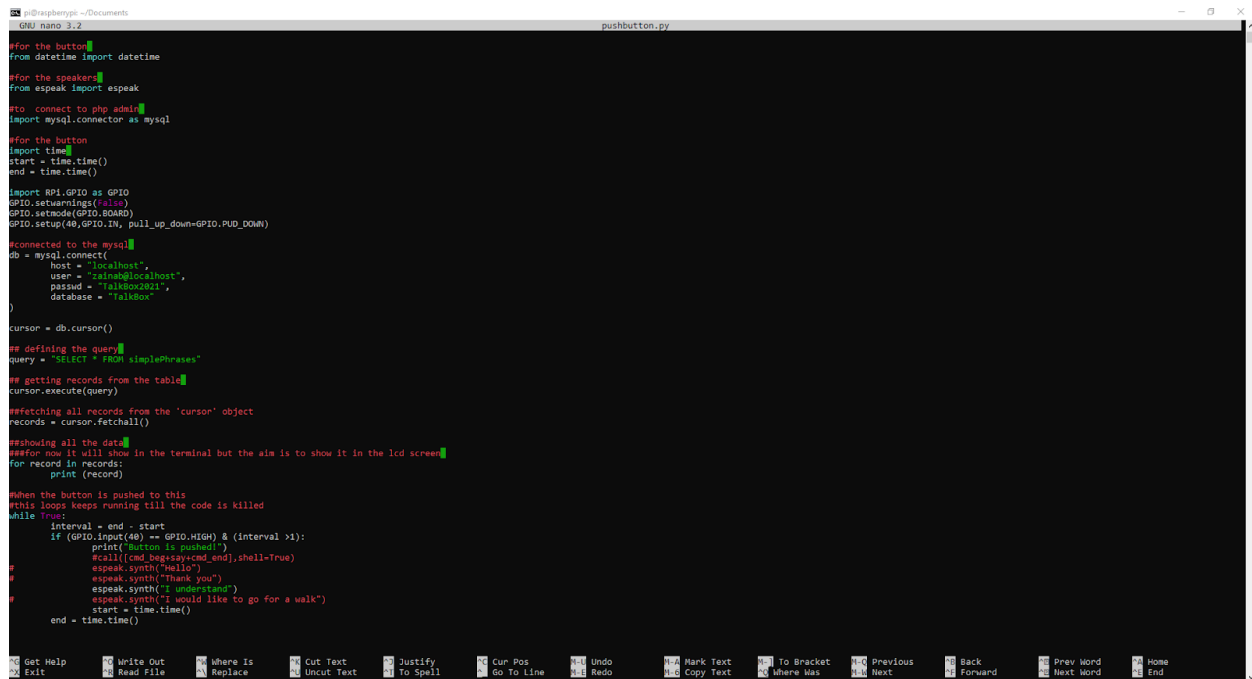


Figure 7: "Simple Phrases" word bank displayed as buttons, ready to be voiced via speaker

3.2.5 Button and Speaker



```
pi@raspberrypi:~/Documents
GNU nano 3.2 pushbutton.py

#for the button
from datetime import datetime

#for the speaker
from espeak import espeak

#to connect to php admin
import mysql.connector as mysql

#for the button
import time
start = time.time()
end = time.time()

import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

#connected to the mysql
db = mysql.connect(
    host = "localhost",
    user = "rain@localhost",
    passwd = "rain@2021",
    database = "rainbox"
)

cursor = db.cursor()

## defining the query
query = "SELECT * FROM simplePhrases"

## getting records from the table
cursor.execute(query)

##fetching all records from the 'cursor' object
records = cursor.fetchall()

##Showing all the data
##for now it will show in the terminal but the aim is to show it in the lcd screen
for record in records:
    print (record)

#when the button is pushed to this
#this loop keeps running till the code is killed
while True:
    interval = end - start
    if (GPIO.input(40) == GPIO.HIGH) & (interval >1):
        print("button is pushed")
        #call cmd to espeak and end, shell=True)
        espeak.synth("hello")
        #
        espeak.synth("Thank you")
        #
        espeak.synth("I understand")
        #
        espeak.synth("I would like to go for a walk")
        start = time.time()
    end = time.time()
```

Figure 8: pushbutton.py script

The python script “pushbutton.py”, which is stored on the Raspberry Pi, connects MySQL and queries from the database and outputs the phrases to espeak. This script allows the team to test text-to-speak functionality. By pushing the button, the script sends data to the espeak.synth() method, and then the phrase is to be said out loud. (Figure 9: button SolidWord design) demonstrates the SolidWorks design to fit the arcade low force push button and mounting strap. (Figure 10: 3D printed button mounted) demonstrates the printed button and wiring soldered to the button to the arcade button. Finally (Figure 11: full assembly of the button) demonstrated all the parts of the button.

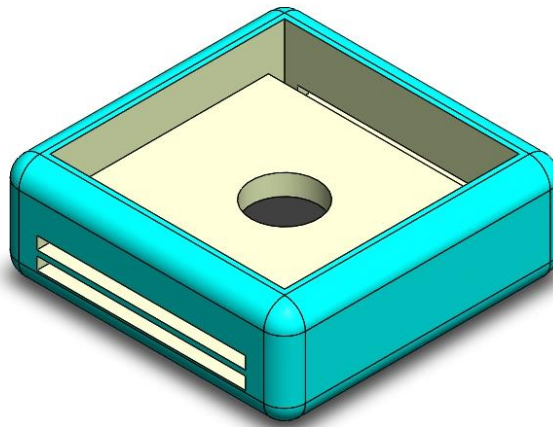


Figure 9: Button SolidWorks Design

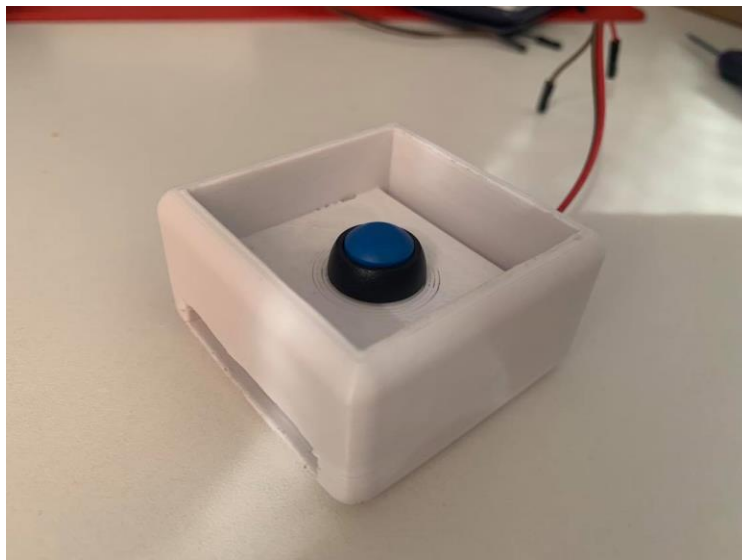


Figure 10: 3D printed and Button mounted

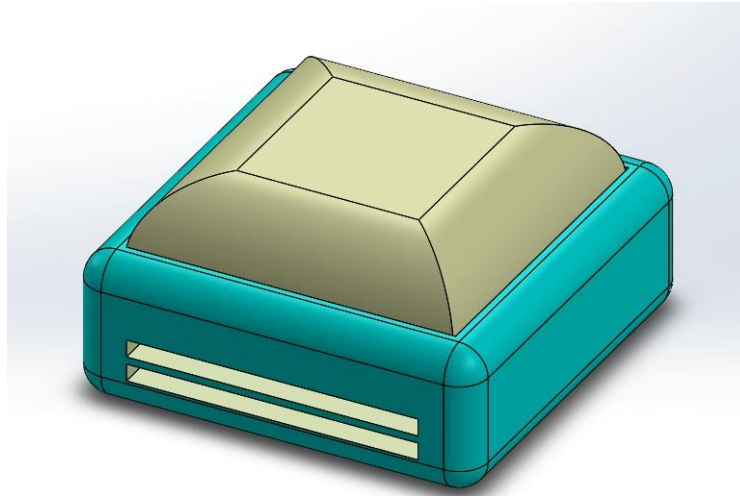


Figure 11: Full assembly of the button

3.2.6 joystick

To create a comfortable grip for the user, a joystick extension was created where the user has more contact area to control the device. A 3D model of the 3-axis joystick that was used in prototype 1, was downloaded from Grabcad and used to create the sketch that would then be used to extrude cut a loft feature of the joystick extension. A cross section of the joystick is shown in (Figure 12: Joystick extension SolidWorks cross section). The joystick extension solidworks files were 3D printed and fitted into the joystick as shown in (Figure 13: 3D printed joystick extension).

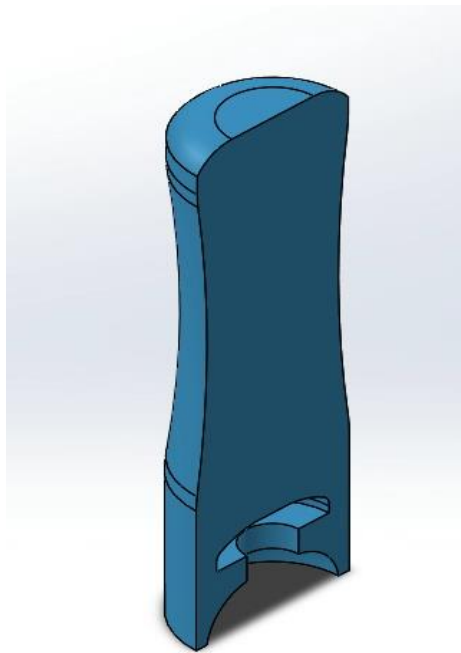


Figure 12: Joystick extension SolidWorks cross section



Figure 13: 3D printed joystick extension

4 Prototype Testing & Evaluation

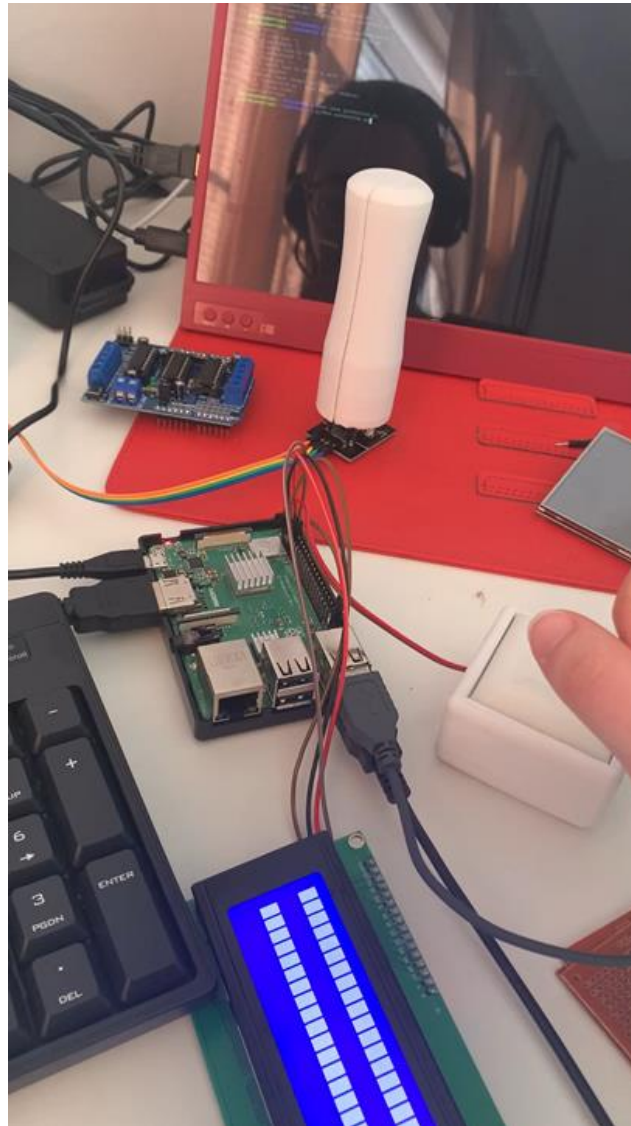


Figure 14: Hardware prototype parts

The test result of the hardware prototype is how the button triggers the talkbox, querying data from MySQL, and then sending commands to the speaker to output phrases. The next iterations of the prototype will use the joystick to scroll as well as select between options, which will eliminate the use of the button. We were able to test the button response time, which is relatively the same for the joystick. The sound quality produced by the speakers were adequate in our prototype testing, however in the next iteration the talkbox sounds will be tested to communicate with a smart home assistant and ensure that the talkbox can produce well enough

sound to awaken smart home assistants. By switching the phpMyAdmin to be hosted on a public server, it will allow the caregiver to access the database remotely which gives the device the ability to easily upgrade. Currently, the joystick and the screen are designed to be mounted on either arm of the wheelchair, however a gooseneck mount for the screen will be used to allow the user to comfortably view the screen from any angle. The weight of the device is the sum of the Raspberry Pi, the screen, and the joystick mount.

In terms of the software prototype, we were able to measure the user friendliness of our interface. We believe in terms of appearance and ease of use for the user, that our current prototype is still not up to par since the font is quite small, therefore it is hard to read. The frame of the GUI also changes sizes from switching between reading small and large text files since the buttons are created in a row, which is quite displeasing. The quality of images displayed on the screen of the user was also evaluated and they were clear and visually pleasing and added a nice effect to our home page. Although we could not obtain a hard value for the image quality, we were able to judge them visually. Unfortunately, due to the small screen size, we were not able to see all the buttons that were created, which is an issue that will be addressed in the future prototype by allowing only 2-3 buttons per page and adding the functionality of going between different pages which would encompass the different options of buttons available to the user.

Table 1: Target Specifications Comparison

#	Metric	Desired Value	Actual Value
1	Button response time	Fast Processing time	1s< - immediate
2	Capability of simple dialogue commands	Processing time	NT
3	Sound produced	>60 dB	NT

4	Connection to Home devices (TV, thermostat, Wi-Fi)	# >1	as many as phrases in the database
5	User friendliness	3	1
6	Quality of images displayed	146 PPI	NT
7	Force required to push button	>4 N	~5N
8	Travel Before Activation	cm	3cm
9	Size of button(s)	6.5 cm ²	4cm
10	Space between button(s)	cm	NA
11	Operating temperature	15-30 °C	10~30°C
12	Languages supported	#1-2	1
13	Emergency procedure	Yes	No
14	Capability of upgrade	Yes	Yes
15	Ability to mount on either side of wheelchair	Wire length	Yes
16	Device comfort	Comfort scale	3
17	Electric Consumption	2-5 W	NT
18	Maximum cost	>= \$100	\$120
19	Screen Size	38 cm	8.89cm
20	Weight	<1000 g	100g
21	Data Storage	16 GB	NT

Table 1: Actual vs Desired values for final target specifications

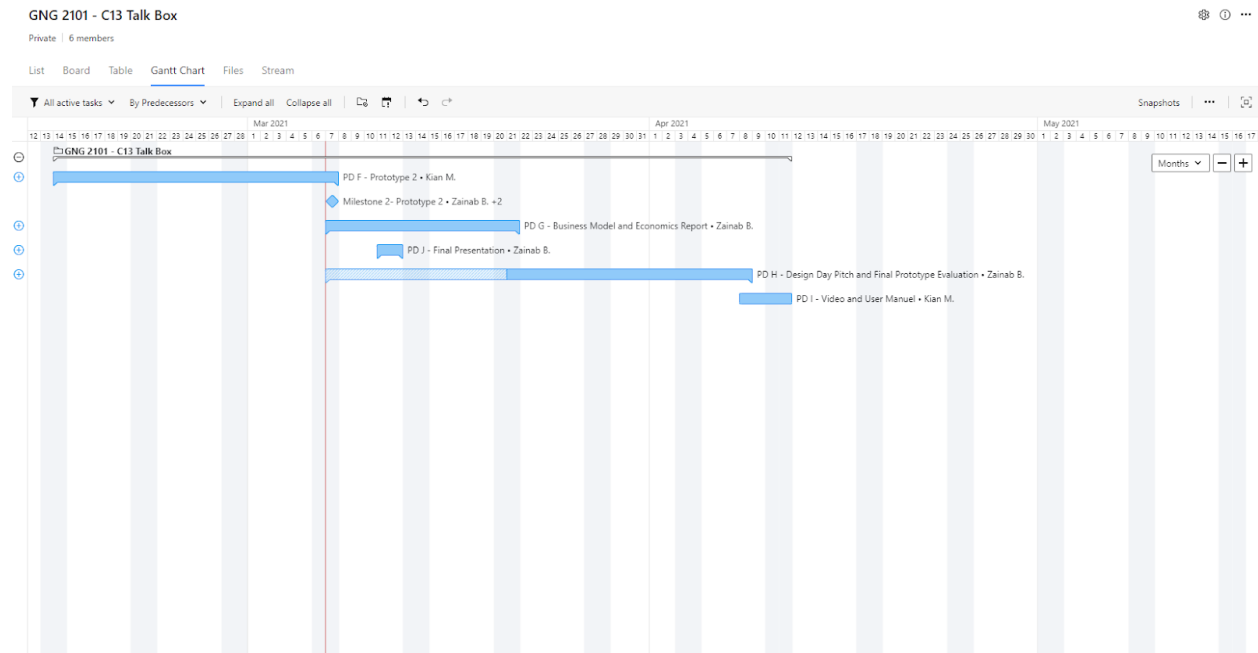
NT = Not tested in this current prototype

5 Conclusions and Recommendations for Future Work

We were successful in being able to implement a basic python software where it offers a simple GUI that has buttons and pictures to go along with it in order to navigate through the distinct categories of phrases. The Chassis for joystick, button, and the Raspberry Pi were reimaged and redesigned to coincide with the client needs and their feedback. There are many tasks that have begun and are work in progress regarding prototype development such as, setting up MySQL and a web page interface for the caretaker to update the python app's data, Test and adjust different fonts and sizes of text and pictures within the python app - which will be showcased on a panel that is also being worked on and is in development. A gooseneck mount is a very viable option which was recommended by the client and inheriting that alongside our mounting design for the microcontroller, will require us to rework our overall design for the wheelchair mount to an extent. Finally, the last aspect of the software prototype, which will advance the project design the most towards completion, is where we implement the selection system, where it doesn't rely on a mouse, but the joystick instead.

APPENDICES

APPENDIX I:



References

Projects.raspberrypi.org, projects.raspberrypi.org/en/projects/raspberry-pi-setting-up.

Emmet. "Setup a Raspberry Pi MYSQL Database." *Pi My Life Up*, 9 Feb. 2021, pimylifeup.com/raspberry-pi-mysql/.

Emmet. "How to Install PHPMyAdmin for the Raspberry Pi." *Pi My Life Up*, 11 Feb. 2021, pimylifeup.com/raspberry-pi-phpmyadmin/.

Hussain, Aamir. "Raspberry Pi 3 Complete Tutorial - Let's Get Started." *Trick i Know*, 20 Apr. 2020, trickiknow.com/raspberry-pi-3-complete-tutorial-2018-lets-get-started/.