

FITNESS TRACKER APP

Group6

Elena Murgia, Paula Calvo, Deborah Kamanya, Nora Rodriguez, Amber Mahmood, Ekta Shah

Table of Contents

Introduction.....	2
Aims and objectives of the project.....	2
Background.....	2
Core functionality	3
User management	3
Main Menu Options	4
Workout Menu	4
Exercise search and storage.....	5
Specifications and Design	5
Functional VS Non-functional Requirements.....	5
Design and Architecture.....	6
Implementation and Execution	7
• Development approaches and team member roles.....	7
• Implementation process	8
Testing and Evaluation	9
Testing strategy	9
Functional and user testing	9
System limitations	10
Conclusion	10

Introduction

This report is organised into several key sections. We begin with an Introduction outlining the goals and objectives of the Fitness Tracker App. The Background section describes the core functionality, including user flows and the application's main features. Specifications and Design cover the functional and non-functional requirements, followed by a breakdown of the application's architecture and design decisions. The Implementation and Execution section discusses the development process, the main challenges encountered, and the tools and technologies used. In Testing and Evaluation, we explain the testing strategy, methods used and system limitations. Finally, the report concludes with a summary of the outcomes and reflections on the development experience.

Aims and objectives of the project

The project aims to develop a user-friendly fitness tracking application that efficiently supports planning, tracking and maintaining a workout routine. The application provides exercise suggestions using an external API, personalisation of the exercises (via user input), and a workout log to track the user's workouts.

The main purposes of the project are:

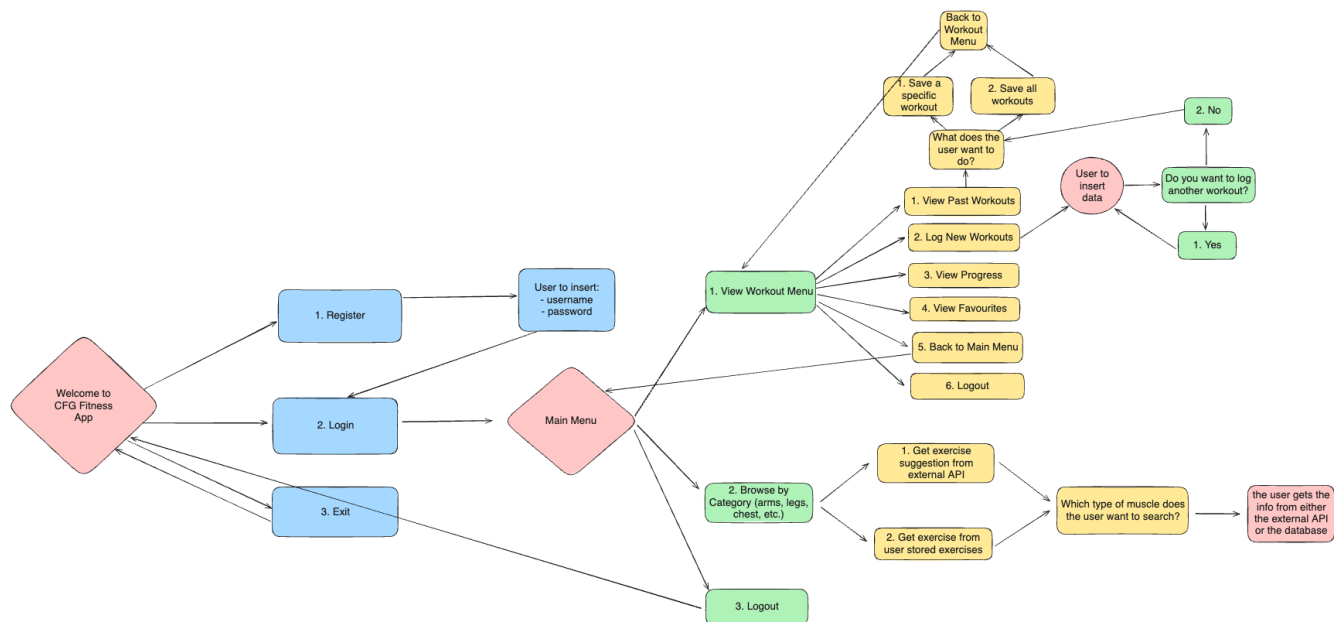
- Integrate an external exercise API (API-Ninjas Exercises) for automated suggestions.
- To allow users to log workouts (MySQL database), including the option to log the exercise sets and view their workout history.
- To allow users to save their workouts to a separate file.
- To enable users to save their chosen exercises from the API and/or create their own.

Background

The Fitness tracker App is designed as a system that allows users to register, log in, browse exercises, log their workouts including the exercise sets and the option to save their workouts to a separate file. The app aims to create a personalised experience where users can easily find exercises, log their workouts, and monitor their improvements while minimising external API calls.

Core functionality

As shown in this diagram:



User management

Users can register with a username and password or log in if they already have an account. After authentication, users are directed to the main menu to navigate different options.

```
00
01 Welcome to the CFG Fitness App
02
03 .. Select a number to choose an option:
04
05 1. Login
06
07 2. Register
08
09 3. Exit
10
11 Choose an option: 1
12
13 Username: paula_C
14
15 Password:
16
17 Login successful. Welcome, paula_C!
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
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
```

Main Menu Options

Users can choose between View the Workout Menu, Browse exercises by category or Logout.

```
MAIN MENU
Select a number to choose an option:
1. View Workout Menu
2. Browse by Category (arms, legs, chest, etc.)
3. Logout
>> 1
```

Workout Menu

Users can view their past workouts, log new ones, track their progress and log out from the Workout Menu.

```
WORKOUT MENU
1. View Past Workouts
2. Log New Workout
3. View Progress
4. View Favourites
5. Go back to the Main Menu
6. Logout
>>
```

Users can choose to View Past Workouts and Log New workouts. Users can also save their workouts to a file.

```
WORKOUT MENU
1. View Past Workouts
2. Log New Workout
3. View Progress
4. View Favourites
5. Go back to the Main Menu
6. Logout
>> 2

Available Exercises:
1. Push-Up (chest, beginner)
2. Squat (legs, beginner)
3. Bicep Curl (arms, intermediate)
4. Plank (core, intermediate)
5. Deadlift (back, advanced)

Enter the number of the exercise you want to log: 1
Duration (in minutes): 20
Any notes or comments that you would like to add? No
Do you want to add sets for this workout? (y/n): n
Workout logged without sets
```

Also, users can go back to the Main Menu or Logout.

Exercise search and storage

Users who want new exercise suggestions when searching for exercises can manually start a new API search or access an existing exercise previously saved in the database. Exercises pulled from the API can be saved into an Exercises table.

Specifications and Design

Functional VS Non-functional Requirements

- **Functional requirements:**
 - Users must be able to register and log in using a unique username and password.
 - Authenticated users must be able to log workouts, view workout history
 - The app must provide exercise suggestions using Ninjas API.
 - Users must be able to save favourite exercises or create custom ones.

- Users should be able to search exercises by muscle group.
 - The system must store all user and workout data in a MySQL database.
 - The terminal interface must provide menu-based navigation for user actions.
- **Non-functional requirements:**
 - Usability: The interface should be intuitive and easy to navigate via the terminal.
 - Reliability: The app should handle errors correctly.
 - Maintainability: Code should be modular and well-documented for future updates or debugging.
 - Scalability: The architecture should allow for future upgrades.

Design and Architecture

- **Terminal interface:**
 Handles user input and the navigation logic.
 Presents the menu to the user(register, login, main menu, workout logging, search, exit).
- **Backend modules:**
 1. User management:
 - Register/login functions
 - Handles session management.
 2. Exercise management:
 - Search in the Exercises table.
 - API integration with the Ninjas API Exercises for external exercise data.
 3. Workout logging:
 - Save the user's workouts to the database.
 4. Diary:
 - Retrieve past workouts based on timeframes.
 5. Data storage:
 - Connects to the database (MySQL)
 - Tables:
 - Users
 - Exercises
 - Exercise sets
 - Workout_Logs

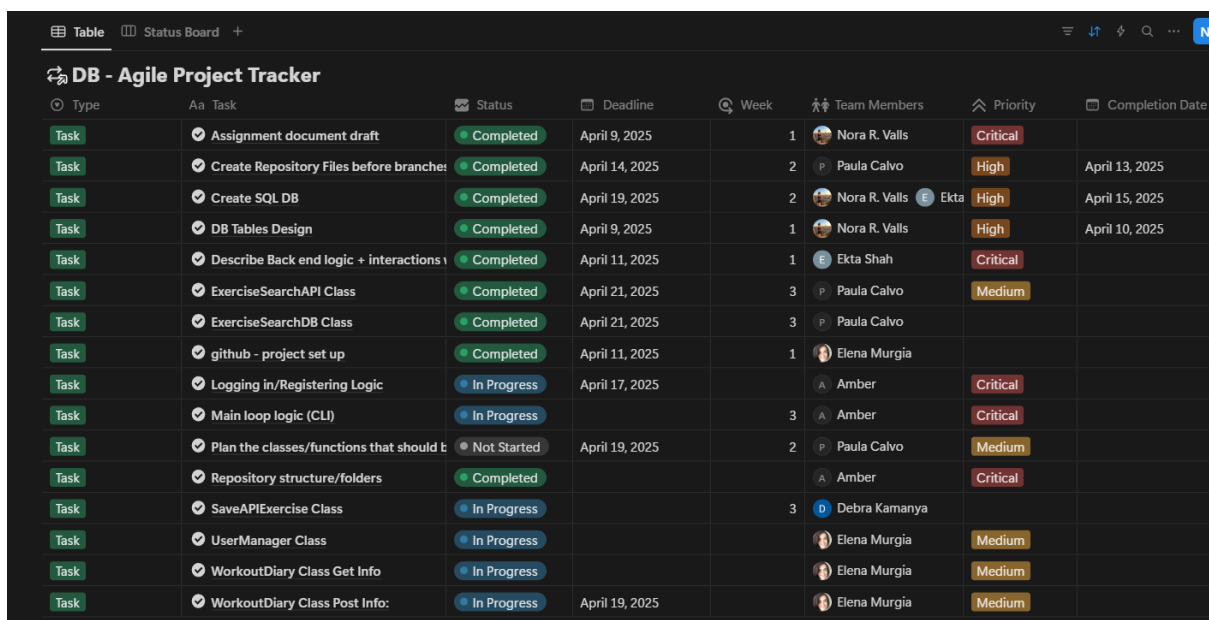
Implementation and Execution

- **Development approaches and team member roles**

We made early design decisions such as incorporating a terminal interface, which API to use (Ninjas API exercises), and the database used (MySQL).

Implementing the Fitness tracker App followed a collaborative and flexible development process, incorporating several Agile development principles throughout the project:

- **Sprint-based planning:** We divided our development into weekly sprints.
- **Task table ownership:** We used a shared table in Notion to track all implementation tasks. Each team member selected the tasks for which they would be responsible and updated their progress through the Slack chat.



Type	Task	Status	Deadline	Week	Team Members	Priority	Completion Date
Task	✓ Assignment document draft	Completed	April 9, 2025	1	Nora R. Valls	Critical	
Task	✓ Create Repository Files before branches	Completed	April 14, 2025	2	Paula Calvo	High	April 13, 2025
Task	✓ Create SQL DB	Completed	April 19, 2025	2	Nora R. Valls, Ekta	High	April 15, 2025
Task	✓ DB Tables Design	Completed	April 9, 2025	1	Nora R. Valls	High	April 10, 2025
Task	✓ Describe Back end logic + interactions	Completed	April 11, 2025	1	Ekta Shah	Critical	
Task	✓ ExerciseSearchAPI Class	Completed	April 21, 2025	3	Paula Calvo	Medium	
Task	✓ ExerciseSearchDB Class	Completed	April 21, 2025	3	Paula Calvo		
Task	✓ github - project set up	Completed	April 11, 2025	1	Elena Murgia		
Task	✓ Logging in/Registering Logic	In Progress	April 17, 2025		Amber	Critical	
Task	✓ Main loop logic (CLI)	In Progress		3	Amber	Critical	
Task	✓ Plan the classes/functions that should be	Not Started	April 19, 2025	2	Paula Calvo	Medium	
Task	✓ Repository structure/folders	Completed			Amber	Critical	
Task	✓ SaveAPIExercise Class	In Progress		3	Debra Kamanya		
Task	✓ UserManager Class	In Progress			Elena Murgia	Medium	
Task	✓ WorkoutDiary Class Get Info	In Progress			Elena Murgia	Medium	
Task	✓ WorkoutDiary Class Post Info:	In Progress	April 19, 2025		Elena Murgia	Medium	

- **Video call retrospectives:** At the end of each sprint, we held a video call using the Gather platform to share updates, demonstrate progress, discuss challenges and agree on the following tasks.

- **Implementation process**

- **Challenges**

- Time management: Due to scheduling conflicts and members' availability, we began coding later than planned.
- External API limitations: Although the Ninjas API was advertised as offering free access, we discovered it was unavailable during certain times of the day unless upgraded to a premium account. This restriction disrupted our testing and development of the exercise search functionality.
- Notion collaboration platform limitations: After two weeks of using Notion to assign tasks and share project documentation, our free access as a team expired. As a result, we transitioned to Google Docs to continue managing and sharing our tasks.
- Testing and debugging: Testing was necessary to ensure no conflicts before merging branches, especially since some code segments were similar. When conflicts did occur, identifying and successfully resolving their sources through debugging required careful attention and collaboration .
- Communication and collaboration: We encountered some collaboration challenges during the development process. Certain team members contributed portions of the codebase at the beginning of the project, making significant changes to the agreed-upon structure, but were later unavailable to explain their implementations. This led to unanticipated structural changes that required time and effort to understand and adapt to. Additionally, a few contributors could not complete their assigned tasks or provide updates, which resulted in the need to revise or remove incomplete code.

- **Achievements**

Despite the challenges faced during development, the team successfully delivered a functional Fitness Tracker App. Some team members demonstrated strong adaptability by restructuring the codebase, resolving merge conflicts, and taking ownership of abandoned tasks.

Collaboration among the remaining members was efficient and solution-focused, allowing the team to overcome setbacks.

The project provided a valuable learning experience in agile teamwork, version control, and problem-solving under pressure.

- **Tools and Libraries**

- **Libraries**

- requests
 - flask
 - unittest
 - unittest.mock
 - pytest
 - json
 - Random
 - Datetime
 - mysql.connector
 - load_dotenv

- **Tools**

- mySQL database
 - API Ninjas exercise
 - Flask
 - PyCharm

Testing and Evaluation

Testing strategy

Our testing strategy combined automated testing using pytest with mock objects via unittest.mock to simulate user inputs and external dependencies.

Debug messages were temporarily included in key code areas to trace logic and identify faulty behaviour during the development.

Exception handling was also integrated to catch and log errors.

Functional and user testing

Functional testing verified that each feature met its requirements. Input handling was validated manually by simulating user interaction in the terminal.

Exceptions were intentionally triggered to ensure that invalid inputs did not cause crashes and that appropriate messages were shown to the user.

System limitations

While most features were tested, some user input cases may not have been exhaustively covered. Debug messages were helpful during development, but were removed in the final steps. Exception handling covers most predictable errors, but unexpected failures may still occur.

Conclusion

The Fitness-tracker App successfully combines local database features with real-time data from an external Exercise API, creating a flexible and user-friendly workout logging experience. Users can securely register and log in, track their workouts with detailed sets and notes, and view progress over time using customisable date filters. The integration with the API Ninjas Exercise API adds a dynamic layer by allowing users to discover new exercises tailored to specific muscle groups, helping to keep their routines engaging and effective. This blend of internal and external resources showcases the app's scalability and potential to evolve into a comprehensive personal fitness assistant. Future improvements may include adding features to collaborate with other users to create a competitive environment, graphical progress charts, and improved validation for user inputs.