



Evidence Gathering Document for SQA Level 8 Professional Developer Award.

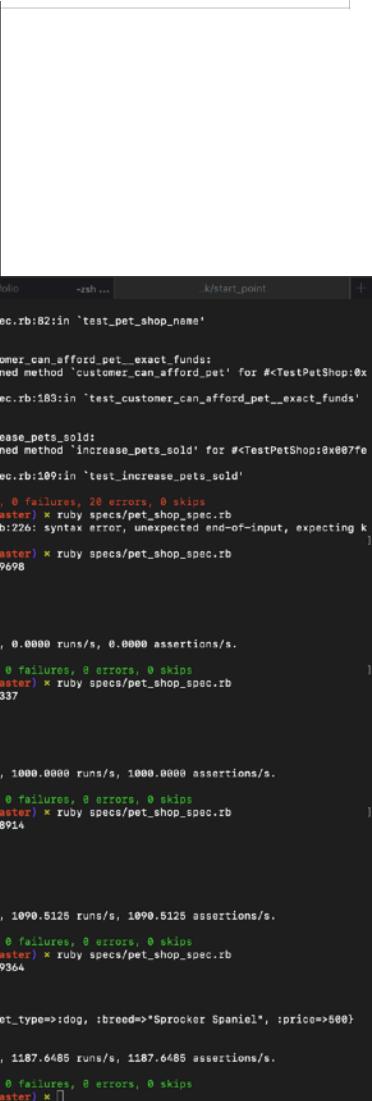
This document is designed for you to present your screenshots and diagrams relevant to the PDA and to also give a short description of what you are showing to clarify understanding for the assessor.

Fill in each point with screenshot or diagram and description of what you are showing.

Each point requires details that cover each element of the Assessment Criteria, along with a brief description of the kind of things you should be showing.

Week 1

Unit	Ref	Evidence
I&T	I.T.6	Demonstrate the use of a hash in a program. Take screenshots of: *A hash in a program *A function that uses the hash *The result of the function running

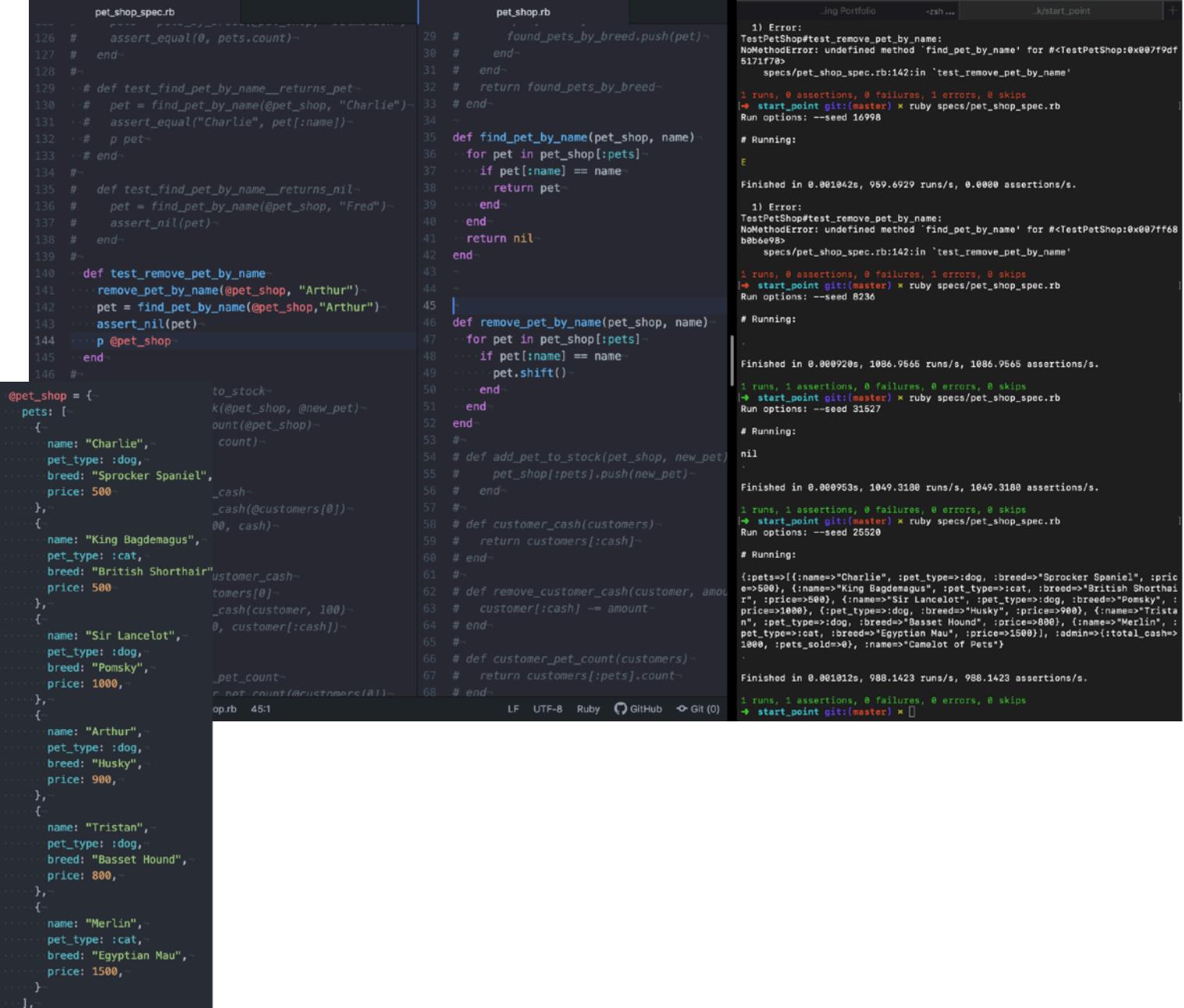
```

@pet_shop = {
  pets: [
    {
      name: "Charlie",
      pet_type: :dog,
      breed: "Sprocker Spaniel",
      price: 500
    }
  ]
}

pet_shop_spec.rb
109 # increase_pets_sold(@pet_shop, 2)
110 # sold = pets_sold(@pet_shop)-
111 # assert_equal(2, sold)-
112 # end-
113 #-
114 # def test_stock_count-
115 # count = stock_count(@pet_shop)-
116 # assert_equal(6, count)-
117 # end-
118 #-
119 # def test_all_pets_by_breed__found-
120 # pets = pets_by_breed(@pet_shop, "British Shor
121 # assert_equal(2, pets.count)-
122 # end-
123 #-
124 # def test_all_pets_by_breed__not_found-
125 # pets = pets_by_breed(@pet_shop, "Dalmation")-
126 # assert_equal(0, pets.count)-
127 # end-
128 #-
129 # def test_find_pet_by_name__returns_pet-
130 # pet = find_pet_by_name(@pet_shop, "Charlie")-
131 # assert_equal("Charlie", pet[:name])-
132 # p pet-
133 # end-
134 #-
135 # def test_find_pet_by_name__returns_nil-
136 # pet = find_pet_by_name(@pet_shop, "Fred")-
137 # assert_nil(pet)-
138 # end-
139 #-
140 # def test_remove_pet_by_name-
141 # remove_pet_by_name(@pet_shop, "Arthur")-
142 # pet = find_pet_by_name(@pet_shop, "Arthur")-
143 # assert_nil(pet)-
144 # end-
145 #-
146 # def test_add_pet_to_stock-
147 # add_pet_to_stock(@pet_shop, @new_pet)-
148 # count = stock_count(@pet_shop)-
149 # def add_pet_to_stock(pet_shop, new_pet)-
150 # for pet in pet_shop[:pets]-
151 # if pet[:name] == name-
152 #   pet.shift()-
153 # end-
154 # end-
155 #-
156 # def increase_pets_sold(pet_shop, number)-
157 # pet_shop[:admin][:pets_sold] += number-
158 # end-
159 #-
160 # def stock_count(pet_shop)-
161 # return pet_shop[:pets].length-
162 # end-
163 #-
164 # def pets_by_breed(pet_shop, breed)-
165 # found_pets_by_breed = []-
166 # for pet in pet_shop[:pets]-
167 # if pet[:breed] == breed-
168 #   found_pets_by_breed.push(pet)-
169 # end-
170 # return found_pets_by_breed-
171 #-
172 # def find_pet_by_name(pet_shop, name)-
173 # for pet in pet_shop[:pets]-
174 # if pet[:name] == name-
175 #   return pet-
176 # end-
177 # return nil-
178 #-
179 # def find_pet_by_name__nil-
180 # pet = find_pet_by_name(@pet_shop, "Fred")-
181 # assert_nil(pet)-
182 #-
183 # def remove_pet_by_name(pet_shop, name)-
184 # for pet in pet_shop[:pets]-
185 # if pet[:name] == name-
186 #   pet.shift()-
187 # end-
188 #-
189 # def add_pet_to_stock(pet_shop, new_pet)-
190 # for pet in pet_shop[:pets]-
191 # if pet[:name] == new_pet[:name]-
192 #   pet[:name] = new_pet[:name]-
193 #   pet[:price] = new_pet[:price]-
194 #   pet[:breed] = new_pet[:breed]-
195 #   pet[:pet_type] = new_pet[:pet_type]-
196 # end-
197 #-
198 # def find_pet_by_name__found-
199 # pet = find_pet_by_name(@pet_shop, "Charlie")-
200 # assert_equal("Charlie", pet[:name])-
201 #-
202 # def stock_count__all_pets-
203 # count = stock_count(@pet_shop)-
204 # assert_equal(12, count)-
205 #-
206 # def increase_pets_sold__all_pets-
207 # increase_pets_sold(@pet_shop, 5)-
208 # count = stock_count(@pet_shop)-
209 # assert_equal(17, count)-
210 #-
211 # def find_pet_by_name__not_found-
212 # pet = find_pet_by_name(@pet_shop, "Dalmation")-
213 # assert_nil(pet)-
214 #-
215 # def remove_pet_by_name__not_found-
216 # remove_pet_by_name(@pet_shop, "Dalmation")-
217 # pet = find_pet_by_name(@pet_shop, "Dalmation")-
218 # assert_nil(pet)-
219 #-
220 # def add_pet_to_stock__not_found-
221 # add_pet_to_stock(@pet_shop, @new_pet)-
222 # pet = find_pet_by_name(@pet_shop, "Charlie")-
223 # assert_nil(pet)-
224 #-
225 # def find_pet_by_name__nil-
226 # pet = find_pet_by_name(@pet_shop, "Fred")-
227 # assert_nil(pet)-
228 #-
229 # def stock_count__empty-
230 # count = stock_count(@pet_shop)-
231 # assert_equal(0, count)-
232 #-
233 # def increase_pets_sold__empty-
234 # increase_pets_sold(@pet_shop, 5)-
235 # count = stock_count(@pet_shop)-
236 # assert_equal(5, count)-
237 #-
238 # def find_pet_by_name__nil-
239 # pet = find_pet_by_name(@pet_shop, "Fred")-
240 # assert_nil(pet)-
241 #-
242 # def remove_pet_by_name__nil-
243 # remove_pet_by_name(@pet_shop, "Fred")-
244 # pet = find_pet_by_name(@pet_shop, "Fred")-
245 # assert_nil(pet)-
246 #-
247 # def add_pet_to_stock__nil-
248 # add_pet_to_stock(@pet_shop, nil)-
249 # pet = find_pet_by_name(@pet_shop, "Fred")-
250 # assert_nil(pet)-
251 #-
252 # def find_pet_by_name__nil-
253 # pet = find_pet_by_name(@pet_shop, "Fred")-
254 # assert_nil(pet)-
255 #-
256 # def stock_count__nil-
257 # count = stock_count(@pet_shop)-
258 # assert_equal(0, count)-
259 #-
260 # def increase_pets_sold__nil-
261 # increase_pets_sold(@pet_shop, 5)-
262 # count = stock_count(@pet_shop)-
263 # assert_equal(5, count)-
264 #-
265 # def find_pet_by_name__nil-
266 # pet = find_pet_by_name(@pet_shop, "Fred")-
267 # assert_nil(pet)-
268 #-
269 # def remove_pet_by_name__nil-
270 # remove_pet_by_name(@pet_shop, nil)-
271 # pet = find_pet_by_name(@pet_shop, "Fred")-
272 # assert_nil(pet)-
273 #-
274 # def add_pet_to_stock__nil-
275 # add_pet_to_stock(@pet_shop, nil)-
276 # pet = find_pet_by_name(@pet_shop, "Fred")-
277 # assert_nil(pet)-
278 #-
279 # def find_pet_by_name__nil-
280 # pet = find_pet_by_name(@pet_shop, "Fred")-
281 # assert_nil(pet)-
282 #-
283 # def stock_count__nil-
284 # count = stock_count(@pet_shop)-
285 # assert_equal(0, count)-
286 #-
287 # def increase_pets_sold__nil-
288 # increase_pets_sold(@pet_shop, 5)-
289 # count = stock_count(@pet_shop)-
290 # assert_equal(5, count)-
291 #-
292 # def find_pet_by_name__nil-
293 # pet = find_pet_by_name(@pet_shop, "Fred")-
294 # assert_nil(pet)-
295 #-
296 # def remove_pet_by_name__nil-
297 # remove_pet_by_name(@pet_shop, nil)-
298 # pet = find_pet_by_name(@pet_shop, "Fred")-
299 # assert_nil(pet)-
300 #-
301 # def add_pet_to_stock__nil-
302 # add_pet_to_stock(@pet_shop, nil)-
303 # pet = find_pet_by_name(@pet_shop, "Fred")-
304 # assert_nil(pet)-
305 #-
306 # def find_pet_by_name__nil-
307 # pet = find_pet_by_name(@pet_shop, "Fred")-
308 # assert_nil(pet)-
309 #-
310 # def stock_count__nil-
311 # count = stock_count(@pet_shop)-
312 # assert_equal(0, count)-
313 #-
314 # def increase_pets_sold__nil-
315 # increase_pets_sold(@pet_shop, 5)-
316 # count = stock_count(@pet_shop)-
317 # assert_equal(5, count)-
318 #-
319 # def find_pet_by_name__nil-
320 # pet = find_pet_by_name(@pet_shop, "Fred")-
321 # assert_nil(pet)-
322 #-
323 # def remove_pet_by_name__nil-
324 # remove_pet_by_name(@pet_shop, nil)-
325 # pet = find_pet_by_name(@pet_shop, "Fred")-
326 # assert_nil(pet)-
327 #-
328 # def add_pet_to_stock__nil-
329 # add_pet_to_stock(@pet_shop, nil)-
330 # pet = find_pet_by_name(@pet_shop, "Fred")-
331 # assert_nil(pet)-
332 #-
333 # def find_pet_by_name__nil-
334 # pet = find_pet_by_name(@pet_shop, "Fred")-
335 # assert_nil(pet)-
336 #-
337 # def stock_count__nil-
338 # count = stock_count(@pet_shop)-
339 # assert_equal(0, count)-
340 #-
341 # def increase_pets_sold__nil-
342 # increase_pets_sold(@pet_shop, 5)-
343 # count = stock_count(@pet_shop)-
344 # assert_equal(5, count)-
345 #-
346 # def find_pet_by_name__nil-
347 # pet = find_pet_by_name(@pet_shop, "Fred")-
348 # assert_nil(pet)-
349 #-
350 # def remove_pet_by_name__nil-
351 # remove_pet_by_name(@pet_shop, nil)-
352 # pet = find_pet_by_name(@pet_shop, "Fred")-
353 # assert_nil(pet)-
354 #-
355 # def add_pet_to_stock__nil-
356 # add_pet_to_stock(@pet_shop, nil)-
357 # pet = find_pet_by_name(@pet_shop, "Fred")-
358 # assert_nil(pet)-
359 #-
360 # def find_pet_by_name__nil-
361 # pet = find_pet_by_name(@pet_shop, "Fred")-
362 # assert_nil(pet)-
363 #-
364 # def stock_count__nil-
365 # count = stock_count(@pet_shop)-
366 # assert_equal(0, count)-
367 #-
368 # def increase_pets_sold__nil-
369 # increase_pets_sold(@pet_shop, 5)-
370 # count = stock_count(@pet_shop)-
371 # assert_equal(5, count)-
372 #-
373 # def find_pet_by_name__nil-
374 # pet = find_pet_by_name(@pet_shop, "Fred")-
375 # assert_nil(pet)-
376 #-
377 # def remove_pet_by_name__nil-
378 # remove_pet_by_name(@pet_shop, nil)-
379 # pet = find_pet_by_name(@pet_shop, "Fred")-
380 # assert_nil(pet)-
381 #-
382 # def add_pet_to_stock__nil-
383 # add_pet_to_stock(@pet_shop, nil)-
384 # pet = find_pet_by_name(@pet_shop, "Fred")-
385 # assert_nil(pet)-
386 #-
387 # def find_pet_by_name__nil-
388 # pet = find_pet_by_name(@pet_shop, "Fred")-
389 # assert_nil(pet)-
390 #-
391 # def stock_count__nil-
392 # count = stock_count(@pet_shop)-
393 # assert_equal(0, count)-
394 #-
395 # def increase_pets_sold__nil-
396 # increase_pets_sold(@pet_shop, 5)-
397 # count = stock_count(@pet_shop)-
398 # assert_equal(5, count)-
399 #-
400 # def find_pet_by_name__nil-
401 # pet = find_pet_by_name(@pet_shop, "Fred")-
402 # assert_nil(pet)-
403 #-
404 # def remove_pet_by_name__nil-
405 # remove_pet_by_name(@pet_shop, nil)-
406 # pet = find_pet_by_name(@pet_shop, "Fred")-
407 # assert_nil(pet)-
408 #-
409 # def add_pet_to_stock__nil-
410 # add_pet_to_stock(@pet_shop, nil)-
411 # pet = find_pet_by_name(@pet_shop, "Fred")-
412 # assert_nil(pet)-
413 #-
414 # def find_pet_by_name__nil-
415 # pet = find_pet_by_name(@pet_shop, "Fred")-
416 # assert_nil(pet)-
417 #-
418 # def stock_count__nil-
419 # count = stock_count(@pet_shop)-
420 # assert_equal(0, count)-
421 #-
422 # def increase_pets_sold__nil-
423 # increase_pets_sold(@pet_shop, 5)-
424 # count = stock_count(@pet_shop)-
425 # assert_equal(5, count)-
426 #-
427 # def find_pet_by_name__nil-
428 # pet = find_pet_by_name(@pet_shop, "Fred")-
429 # assert_nil(pet)-
430 #-
431 # def remove_pet_by_name__nil-
432 # remove_pet_by_name(@pet_shop, nil)-
433 # pet = find_pet_by_name(@pet_shop, "Fred")-
434 # assert_nil(pet)-
435 #-
436 # def add_pet_to_stock__nil-
437 # add_pet_to_stock(@pet_shop, nil)-
438 # pet = find_pet_by_name(@pet_shop, "Fred")-
439 # assert_nil(pet)-
440 #-
441 # def find_pet_by_name__nil-
442 # pet = find_pet_by_name(@pet_shop, "Fred")-
443 # assert_nil(pet)-
444 #-
445 # def stock_count__nil-
446 # count = stock_count(@pet_shop)-
447 # assert_equal(0, count)-
448 #-
449 # def increase_pets_sold__nil-
450 # increase_pets_sold(@pet_shop, 5)-
451 # count = stock_count(@pet_shop)-
452 # assert_equal(5, count)-
453 #-
454 # def find_pet_by_name__nil-
455 # pet = find_pet_by_name(@pet_shop, "Fred")-
456 # assert_nil(pet)-
457 #-
458 # def remove_pet_by_name__nil-
459 # remove_pet_by_name(@pet_shop, nil)-
460 # pet = find_pet_by_name(@pet_shop, "Fred")-
461 # assert_nil(pet)-
462 #-
463 # def add_pet_to_stock__nil-
464 # add_pet_to_stock(@pet_shop, nil)-
465 # pet = find_pet_by_name(@pet_shop, "Fred")-
466 # assert_nil(pet)-
467 #-
468 # def find_pet_by_name__nil-
469 # pet = find_pet_by_name(@pet_shop, "Fred")-
470 # assert_nil(pet)-
471 #-
472 # def stock_count__nil-
473 # count = stock_count(@pet_shop)-
474 # assert_equal(0, count)-
475 #-
476 # def increase_pets_sold__nil-
477 # increase_pets_sold(@pet_shop, 5)-
478 # count = stock_count(@pet_shop)-
479 # assert_equal(5, count)-
480 #-
481 # def find_pet_by_name__nil-
482 # pet = find_pet_by_name(@pet_shop, "Fred")-
483 # assert_nil(pet)-
484 #-
485 # def remove_pet_by_name__nil-
486 # remove_pet_by_name(@pet_shop, nil)-
487 # pet = find_pet_by_name(@pet_shop, "Fred")-
488 # assert_nil(pet)-
489 #-
490 # def add_pet_to_stock__nil-
491 # add_pet_to_stock(@pet_shop, nil)-
492 # pet = find_pet_by_name(@pet_shop, "Fred")-
493 # assert_nil(pet)-
494 #-
495 # def find_pet_by_name__nil-
496 # pet = find_pet_by_name(@pet_shop, "Fred")-
497 # assert_nil(pet)-
498 #-
499 # def stock_count__nil-
500 # count = stock_count(@pet_shop)-
501 # assert_equal(0, count)-
502 #-
503 # def increase_pets_sold__nil-
504 # increase_pets_sold(@pet_shop, 5)-
505 # count = stock_count(@pet_shop)-
506 # assert_equal(5, count)-
507 #-
508 # def find_pet_by_name__nil-
509 # pet = find_pet_by_name(@pet_shop, "Fred")-
510 # assert_nil(pet)-
511 #-
512 # def remove_pet_by_name__nil-
513 # remove_pet_by_name(@pet_shop, nil)-
514 # pet = find_pet_by_name(@pet_shop, "Fred")-
515 # assert_nil(pet)-
516 #-
517 # def add_pet_to_stock__nil-
518 # add_pet_to_stock(@pet_shop, nil)-
519 # pet = find_pet_by_name(@pet_shop, "Fred")-
520 # assert_nil(pet)-
521 #-
522 # def find_pet_by_name__nil-
523 # pet = find_pet_by_name(@pet_shop, "Fred")-
524 # assert_nil(pet)-
525 #-
526 # def stock_count__nil-
527 # count = stock_count(@pet_shop)-
528 # assert_equal(0, count)-
529 #-
530 # def increase_pets_sold__nil-
531 # increase_pets_sold(@pet_shop, 5)-
532 # count = stock_count(@pet_shop)-
533 # assert_equal(5, count)-
534 #-
535 # def find_pet_by_name__nil-
536 # pet = find_pet_by_name(@pet_shop, "Fred")-
537 # assert_nil(pet)-
538 #-
539 # def remove_pet_by_name__nil-
540 # remove_pet_by_name(@pet_shop, nil)-
541 # pet = find_pet_by_name(@pet_shop, "Fred")-
542 # assert_nil(pet)-
543 #-
544 # def add_pet_to_stock__nil-
545 # add_pet_to_stock(@pet_shop, nil)-
546 # pet = find_pet_by_name(@pet_shop, "Fred")-
547 # assert_nil(pet)-
548 #-
549 # def find_pet_by_name__nil-
550 # pet = find_pet_by_name(@pet_shop, "Fred")-
551 # assert_nil(pet)-
552 #-
553 # def stock_count__nil-
554 # count = stock_count(@pet_shop)-
555 # assert_equal(0, count)-
556 #-
557 # def increase_pets_sold__nil-
558 # increase_pets_sold(@pet_shop, 5)-
559 # count = stock_count(@pet_shop)-
560 # assert_equal(5, count)-
561 #-
562 # def find_pet_by_name__nil-
563 # pet = find_pet_by_name(@pet_shop, "Fred")-
564 # assert_nil(pet)-
565 #-
566 # def remove_pet_by_name__nil-
567 # remove_pet_by_name(@pet_shop, nil)-
568 # pet = find_pet_by_name(@pet_shop, "Fred")-
569 # assert_nil(pet)-
570 #-
571 # def add_pet_to_stock__nil-
572 # add_pet_to_stock(@pet_shop, nil)-
573 # pet = find_pet_by_name(@pet_shop, "Fred")-
574 # assert_nil(pet)-
575 #-
576 # def find_pet_by_name__nil-
577 # pet = find_pet_by_name(@pet_shop, "Fred")-
578 # assert_nil(pet)-
579 #-
580 # def stock_count__nil-
581 # count = stock_count(@pet_shop)-
582 # assert_equal(0, count)-
583 #-
584 # def increase_pets_sold__nil-
585 # increase_pets_sold(@pet_shop, 5)-
586 # count = stock_count(@pet_shop)-
587 # assert_equal(5, count)-
588 #-
589 # def find_pet_by_name__nil-
590 # pet = find_pet_by_name(@pet_shop, "Fred")-
591 # assert_nil(pet)-
592 #-
593 # def remove_pet_by_name__nil-
594 # remove_pet_by_name(@pet_shop, nil)-
595 # pet = find_pet_by_name(@pet_shop, "Fred")-
596 # assert_nil(pet)-
597 #-
598 # def add_pet_to_stock__nil-
599 # add_pet_to_stock(@pet_shop, nil)-
600 # pet = find_pet_by_name(@pet_shop, "Fred")-
601 # assert_nil(pet)-
602 #-
603 # def find_pet_by_name__nil-
604 # pet = find_pet_by_name(@pet_shop, "Fred")-
605 # assert_nil(pet)-
606 #-
607 # def stock_count__nil-
608 # count = stock_count(@pet_shop)-
609 # assert_equal(0, count)-
610 #-
611 # def increase_pets_sold__nil-
612 # increase_pets_sold(@pet_shop, 5)-
613 # count = stock_count(@pet_shop)-
614 # assert_equal(5, count)-
615 #-
616 # def find_pet_by_name__nil-
617 # pet = find_pet_by_name(@pet_shop, "Fred")-
618 # assert_nil(pet)-
619 #-
620 # def remove_pet_by_name__nil-
621 # remove_pet_by_name(@pet_shop, nil)-
622 # pet = find_pet_by_name(@pet_shop, "Fred")-
623 # assert_nil(pet)-
624 #-
625 # def add_pet_to_stock__nil-
626 # add_pet_to_stock(@pet_shop, nil)-
627 # pet = find_pet_by_name(@pet_shop, "Fred")-
628 # assert_nil(pet)-
629 #-
630 # def find_pet_by_name__nil-
631 # pet = find_pet_by_name(@pet_shop, "Fred")-
632 # assert_nil(pet)-
633 #-
634 # def stock_count__nil-
635 # count = stock_count(@pet_shop)-
636 # assert_equal(0, count)-
637 #-
638 # def increase_pets_sold__nil-
639 # increase_pets_sold(@pet_shop, 5)-
640 # count = stock_count(@pet_shop)-
641 # assert_equal(5, count)-
642 #-
643 # def find_pet_by_name__nil-
644 # pet = find_pet_by_name(@pet_shop, "Fred")-
645 # assert_nil(pet)-
646 #-
647 # def remove_pet_by_name__nil-
648 # remove_pet_by_name(@pet_shop, nil)-
649 # pet = find_pet_by_name(@pet_shop, "Fred")-
650 # assert_nil(pet)-
651 #-
652 # def add_pet_to_stock__nil-
653 # add_pet_to_stock(@pet_shop, nil)-
654 # pet = find_pet_by_name(@pet_shop, "Fred")-
655 # assert_nil(pet)-
656 #-
657 # def find_pet_by_name__nil-
658 # pet = find_pet_by_name(@pet_shop, "Fred")-
659 # assert_nil(pet)-
660 #-
661 # def stock_count__nil-
662 # count = stock_count(@pet_shop)-
663 # assert_equal(0, count)-
664 #-
665 # def increase_pets_sold__nil-
666 # increase_pets_sold(@pet_shop, 5)-
667 # count = stock_count(@pet_shop)-
668 # assert_equal(5, count)-
669 #-
670 # def find_pet_by_name__nil-
671 # pet = find_pet_by_name(@pet_shop, "Fred")-
672 # assert_nil(pet)-
673 #-
674 # def remove_pet_by_name__nil-
675 # remove_pet_by_name(@pet_shop, nil)-
676 # pet = find_pet_by_name(@pet_shop, "Fred")-
677 # assert_nil(pet)-
678 #-
679 # def add_pet_to_stock__nil-
680 # add_pet_to_stock(@pet_shop, nil)-
681 # pet = find_pet_by_name(@pet_shop, "Fred")-
682 # assert_nil(pet)-
683 #-
684 # def find_pet_by_name__nil-
685 # pet = find_pet_by_name(@pet_shop, "Fred")-
686 # assert_nil(pet)-
687 #-
688 # def stock_count__nil-
689 # count = stock_count(@pet_shop)-
690 # assert_equal(0, count)-
691 #-
692 # def increase_pets_sold__nil-
693 # increase_pets_sold(@pet_shop, 5)-
694 # count = stock_count(@pet_shop)-
695 # assert_equal(5, count)-
696 #-
697 # def find_pet_by_name__nil-
698 # pet = find_pet_by_name(@pet_shop, "Fred")-
699 # assert_nil(pet)-
700 #-
701 # def remove_pet_by_name__nil-
702 # remove_pet_by_name(@pet_shop, nil)-
703 # pet = find_pet_by_name(@pet_shop, "Fred")-
704 # assert_nil(pet)-
705 #-
706 # def add_pet_to_stock__nil-
707 # add_pet_to_stock(@pet_shop, nil)-
708 # pet = find_pet_by_name(@pet_shop, "Fred")-
709 # assert_nil(pet)-
710 #-
711 # def find_pet_by_name__nil-
712 # pet = find_pet_by_name(@pet_shop, "Fred")-
713 # assert_nil(pet)-
714 #-
715 # def stock_count__nil-
716 # count = stock_count(@pet_shop)-
717 # assert_equal(0, count)-
718 #-
719 # def increase_pets_sold__nil-
720 # increase_pets_sold(@pet_shop, 5)-
721 # count = stock_count(@pet_shop)-
722 # assert_equal(5, count)-
723 #-
724 # def find_pet_by_name__nil-
725 # pet = find_pet_by_name(@pet_shop, "Fred")-
726 # assert_nil(pet)-
727 #-
728 # def remove_pet_by_name__nil-
729 # remove_pet_by_name(@pet_shop, nil)-
730 # pet = find_pet_by_name(@pet_shop, "Fred")-
731 # assert_nil(pet)-
732 #-
733 # def add_pet_to_stock__nil-
734 # add_pet_to_stock(@pet_shop, nil)-
735 # pet = find_pet_by_name(@pet_shop, "Fred")-
736 # assert_nil(pet)-
737 #-
738 # def find_pet_by_name__nil-
739 # pet = find_pet_by_name(@pet_shop, "Fred")-
740 # assert_nil(pet)-
741 #-
742 # def stock_count__nil-
743 # count = stock_count(@pet_shop)-
744 # assert_equal(0, count)-
745 #-
746 # def increase_pets_sold__nil-
747 # increase_pets_sold(@pet_shop, 5)-
748 # count = stock_count(@pet_shop)-
749 # assert_equal(5, count)-
750 #-
751 # def find_pet_by_name__nil-
752 # pet = find_pet_by_name(@pet_shop, "Fred")-
753 # assert_nil(pet)-
754 #-
755 # def remove_pet_by_name__nil-
756 # remove_pet_by_name(@pet_shop, nil)-
757 # pet = find_pet_by_name(@pet_shop, "Fred")-
758 # assert_nil(pet)-
759 #-
760 # def add_pet_to_stock__nil-
761 # add_pet_to_stock(@pet_shop, nil)-
762 # pet = find_pet_by_name(@pet_shop, "Fred")-
763 # assert_nil(pet)-
764 #-
765 # def find_pet_by_name__nil-
766 # pet = find_pet_by_name(@pet_shop, "Fred")-
767 # assert_nil(pet)-
768 #-
769 # def stock_count__nil-
770 # count = stock_count(@pet_shop)-
771 # assert_equal(0, count)-
772 #-
773 # def increase_pets_sold__nil-
774 # increase_pets_sold(@pet_shop, 5)-
775 # count = stock_count(@pet_shop)-
776 # assert_equal(5, count)-
777 #-
778 # def find_pet_by_name__nil-
779 # pet = find_pet_by_name(@pet_shop, "Fred")-
780 # assert_nil(pet)-
781 #-
782 # def remove_pet_by_name__nil-
783 # remove_pet_by_name(@pet_shop, nil)-
784 # pet = find_pet_by_name(@pet_shop, "Fred")-
785 # assert_nil(pet)-
786 #-
787 # def add_pet_to_stock__nil-
788 # add_pet_to_stock(@pet_shop, nil)-
789 # pet = find_pet_by_name(@pet_shop, "Fred")-
790 # assert_nil(pet)-
791 #-
792 # def find_pet_by_name__nil-
793 # pet = find_pet_by_name(@pet_shop, "Fred")-
794 # assert_nil(pet)-
795 #-
796 # def stock_count__nil-
797 # count = stock_count(@pet_shop)-
798 # assert_equal(0, count)-
799 #-
800 # def increase_pets_sold__nil-
801 # increase_pets_sold(@pet_shop, 5)-
802 # count = stock_count(@pet_shop)-
803 # assert_equal(5, count)-
804 #-
805 # def find_pet_by_name__nil-
806 # pet = find_pet_by_name(@pet_shop, "Fred")-
807 # assert_nil(pet)-
808 #-
809 # def remove_pet_by_name__nil-
810 # remove_pet_by_name(@pet_shop, nil)-
811 # pet = find_pet_by_name(@pet_shop, "Fred")-
812 # assert_nil(pet)-
813 #-
814 # def add_pet_to_stock__nil-
815 # add_pet_to_stock(@pet_shop, nil)-
816 # pet = find_pet_by_name(@pet_shop, "Fred")-
817 # assert_nil(pet)-
818 #-
819 # def find_pet_by_name__nil-
820 # pet = find_pet_by_name(@pet_shop, "Fred")-
821 # assert_nil(pet)-
822 #-
823 # def stock_count__nil-
824 # count = stock_count(@pet_shop)-
825 # assert_equal(0, count)-
826 #-
827 # def increase_pets_sold__nil-
828 # increase_pets_sold(@pet_shop, 5)-
829 # count = stock_count(@pet_shop)-
830 # assert_equal(5, count)-
831 #-
832 # def find_pet_by_name__nil-
833 # pet = find_pet_by_name(@pet_shop, "Fred")-
834 # assert_nil(pet)-
835 #-
836 # def remove_pet_by_name__nil-
837 # remove_pet_by_name(@pet_shop, nil)-
838 # pet = find_pet_by_name(@pet_shop, "Fred")-
839 # assert_nil(pet)-
840 #-
841 # def add_pet_to_stock__nil-
842 # add_pet_to_stock(@pet_shop, nil)-
843 # pet = find_pet_by_name(@pet_shop, "Fred")-
844 # assert_nil(pet)-
845 #-
846 # def find_pet_by_name__nil-
847 # pet = find_pet_by_name(@pet_shop, "Fred")-
848 # assert_nil(pet)-
849 #-
850 # def stock_count__nil-
851 # count = stock_count(@pet_shop)-
852 # assert_equal(0, count)-
853 #-
854 # def increase_pets_sold__nil-
855 # increase_pets_sold(@pet_shop, 5)-
856 # count = stock_count(@pet_shop)-
857 # assert_equal(5, count)-
858 #-
859 # def find_pet_by_name__nil-
860 # pet = find_pet_by_name(@pet_shop, "Fred")-
861 # assert_nil(pet)-
862 #-
863 # def remove_pet_by_name__nil-
864 # remove_pet_by_name(@pet_shop, nil)-
865 # pet = find_pet_by_name(@pet_shop, "Fred")-
866 # assert_nil(pet)-
867 #-
868 # def add_pet_to_stock__nil-
869 # add_pet_to_stock(@pet_shop, nil)-
870 # pet = find_pet_by_name(@pet_shop, "Fred")-
871 # assert_nil(pet)-
872 #-
873 # def find_pet_by_name__nil-
874 # pet = find_pet_by_name(@pet_shop, "Fred")-
875 # assert_nil(pet)-
876 #-
877 # def stock_count__nil-
878 # count = stock_count(@pet_shop)-
879 # assert_equal(0, count)-
880 #-
881 # def increase_pets_sold__nil-
882 # increase_pets_sold(@pet_shop, 5)-
883 # count = stock_count(@pet_shop)-
884 # assert_equal(5, count)-
885 #-
886 # def find_pet_by_name__nil-
887 # pet = find_pet_by_name(@pet_shop, "Fred")-
888 # assert_nil(pet)-
889 #-
890 # def remove_pet_by_name__nil-
891 # remove_pet_by_name(@pet_shop, nil)-
892 # pet = find_pet_by_name(@pet_shop, "Fred")-
893 # assert_nil(pet)-
894 #-
895 # def add_pet_to_stock__nil-
896 # add_pet_to_stock(@pet_shop, nil)-
897 # pet = find_pet_by_name(@pet_shop, "Fred")-
898 # assert_nil(pet)-
899 #-
900 # def find_pet_by_name__nil-
901 # pet = find_pet_by_name(@pet_shop, "Fred")-
902 # assert_nil(pet)-
903 #-
904 # def stock_count__nil-
905 # count = stock_count(@pet_shop)-
906 # assert_equal(0, count)-
907 #-
908 # def increase_pets_sold__nil-
909 # increase_pets_sold(@pet_shop, 5)-
910 # count = stock_count(@pet_shop)-
911 # assert_equal(5, count)-
912 #-
913 # def find_pet_by_name__nil-
914 # pet = find_pet_by_name(@pet_shop, "Fred")-
915 # assert_nil(pet)-
916 #-
917 # def remove_pet_by_name__nil-
918 # remove_pet_by_name(@pet_shop, nil)-
919 # pet = find_pet_by_name(@pet_shop, "Fred")-
920 # assert_nil(pet)-
921 #-
922 # def add_pet_to_stock__nil-
923 # add_pet_to_stock(@pet_shop, nil)-
924 # pet = find_pet_by_name(@pet_shop, "Fred")-
925 # assert_nil(pet)-
926 #-
927 # def find_pet_by_name__nil-
928 # pet = find_pet_by_name(@pet_shop, "Fred")-
929 # assert_nil(pet)-
930 #-
931 # def stock_count__nil-
932 # count = stock_count(@pet_shop)-
933 # assert_equal(0, count)-
934 #-
935 # def increase_pets_sold__nil-
936 # increase_pets_sold(@pet_shop, 5)-
937 # count = stock_count(@pet_shop)-
938 # assert_equal(5, count)-
939 #-
940 # def find_pet_by_name__nil-
941 # pet = find_pet_by_name(@pet_shop, "Fred")-
942 # assert_nil(pet)-
943 #-
944 # def remove_pet_by_name__nil-
945 # remove_pet_by_name(@pet_shop, nil)-
946 # pet = find_pet_by_name(@pet_shop, "Fred")-
947 # assert_nil(pet)-
948 #-
949 # def add_pet_to_stock__nil-
950 # add_pet_to_stock(@pet_shop, nil)-
951 # pet = find_pet_by_name(@pet_shop, "Fred")-
952 # assert_nil(pet)-
953 #-
954 # def find_pet_by_name__nil-
955 # pet = find_pet_by_name(@pet_shop, "Fred")-
956 # assert_nil(pet)-
957 #-
958 # def stock_count__nil-
959 # count = stock_count(@pet_shop)-
960 # assert_equal(0, count)-
961 #-
962 # def increase_pets_sold__nil-
963 # increase_pets_sold(@pet_shop, 5)-
964 # count = stock_count(@pet_shop)-
965 # assert_equal(5, count)-
966 #-
967 # def find_pet_by_name__nil-
968 # pet = find_pet_by_name(@pet_shop, "Fred")-
969 # assert_nil(pet)-
970 #-
971 # def remove_pet_by_name__nil-
972 # remove_pet_by_name(@pet_shop, nil)-
973 # pet = find_pet_by_name(@pet_shop, "Fred")-
974 # assert_nil(pet)-
975 #-
976 # def add_pet_to_stock__nil-
977 # add_pet_to_stock(@pet_shop, nil)-
978 # pet = find_pet_by_name(@pet_shop, "Fred")-
979 # assert_nil(pet)-
980 #-
981 # def find_pet_by_name__nil-
982 # pet = find_pet_by_name(@pet_shop, "Fred")-
983 # assert_nil(pet)-
984 #-
985 # def stock_count__nil-
986 # count = stock_count(@pet_shop)-
987 # assert_equal(0, count)-
988 #-
989 # def increase_pets_sold__nil-
990 # increase_pets_sold(@pet_shop, 5)-
991 # count = stock_count(@pet_shop)-
992 # assert_equal(5, count)-
993 #-
994 # def find_pet_by_name__nil-
995 # pet = find_pet_by_name(@pet_shop, "Fred")-
996 # assert_nil(pet)-
997 #-
998 # def remove_pet_by_name__nil-
999 # remove_pet_by_name(@pet_shop, nil)-
1000 # pet = find_pet_by_name(@pet_shop, "Fred")-
1001 # assert_nil(pet)-
1002 #-
1003 # def add_pet_to_stock__nil-
1004 # add_pet_to_stock(@pet_shop, nil)-
1005 # pet = find_pet_by_name(@pet_shop, "Fred")-
1006 # assert_nil(pet)-
1007 #-
1008 # def find_pet_by_name__nil-
1009 # pet = find_pet_by_name(@pet_shop, "Fred")-
1010 # assert_nil(pet)-
1011 #-
1012 # def stock_count__nil-
1013 # count = stock_count(@pet_shop)-
1014 # assert_equal(0, count)-
1015 #-
1016 # def increase_pets_sold__nil-
1017 # increase_pets_sold(@pet_shop, 5)-
1018 # count = stock_count(@pet_shop)-
1019 # assert_equal(5, count)-
1020 #-
1021 # def find_pet_by_name__nil-
1022 # pet = find_pet_by_name(@pet_shop, "Fred")-
1023 #
```

Week 2

Unit	Ref	Evidence
I&T	I.T.5	Demonstrate the use of an array in a program. Take screenshots of: *An array in a program *A function that uses the array *The result of the function running



The left-hand screenshot shows an array (an array of hashes) being used for the methods. In the large screenshot you can see a function and its test. This function removes a pet from the array. In the terminal you can see that the test passes (bottom of terminal) and the results have also been printed - Arthur the dog is no longer present in the list.

```
pet_shop_spec.rb
```

```

126 #     assert_equal(0, pets.count)
127 #   end
128 #
129 - def test_find_pet_by_name_returns_pet-
130 -   # pet = find_pet_by_name(@pet_shop, "Charlie")
131 -   # assert_equal("Charlie", pet[:name])
132 -   p pet
133 -   # end
134 #
135 - def test_find_pet_by_name_returns_nil-
136 -   # pet = find_pet_by_name(@pet_shop, "Fred")
137 -   # assert_nil(pet)
138 -   # end
139 #
140 - def test_remove_pet_by_name
141 -   remove_pet_by_name(@pet_shop, "Arthur")
142 -   pet = find_pet_by_name(@pet_shop, "Arthur")
143 -   assert_nil(pet)
144 -   p @pet_shop
145 - end
146 #

```

```

@pet_shop = {
  pets: [
    { name: "Charlie", pet_type: :dog, breed: "Sprocker Spaniel", price: 500 },
    { name: "King Bagdemagus", pet_type: :cat, breed: "British Shorthair", price: 500 },
    { name: "Sir Lancelot", pet_type: :dog, breed: "Pomsky", price: 1000 },
    { name: "Arthur", pet_type: :dog, breed: "Husky", price: 900 },
    { name: "Tristan", pet_type: :dog, breed: "Basset Hound", price: 800 },
    { name: "Merlin", pet_type: :cat, breed: "Egyptian Mau", price: 1500 }
  ],
  to_stock: k(@pet_shop, @new_pet) -> count(@pet_shop) -> count
}

```

```

pet_shop.rb
```

```

29 #     found_pets_by_breed.push(pet)
30 #   end
31 # end
32 # return found_pets_by_breed
33 # end
34 #
35 def find_pet_by_name(pet_shop, name)
36   for pet in pet_shop[:pets]
37     if pet[:name] == name
38       return pet
39     end
40   end
41   return nil
42 end
43 #
44 #
45 def remove_pet_by_name(pet_shop, name)
46   for pet in pet_shop[:pets]
47     if pet[:name] == name
48       pet.shift()
49     end
50   end
51 end
52 #
53 #
54 def add_pet_to_stock(pet_shop, new_pet)
55   pet_shop[:pets].push(new_pet)
56 end
57 #
58 def customer_cash(customers)
59   return customers[:cash]
60 end
61 #
62 def remove_customer_cash(customer, amount)
63   customer[:cash] -= amount
64 end
65 #
66 def customer_pet_count(customers)
67   return customers[:pets].count
68 end

```

```

op.rb 45:1
```

```

..Jng Portfolio -zsh ... .k/start_point
1) Error:
TestPetShop#test_remove_pet_by_name:
NoMethodError: undefined method `find_pet_by_name' for #<TestPetShop:0x007f9df5171f78>
specs/pet_shop_spec.rb:142:in `test_remove_pet_by_name'
1 runs, 0 assertions, 0 failures, 1 errors, 0 skips
→ start_point git:(master) ✘ ruby specs/pet_shop_spec.rb
Run options: --seed 16998

# Running:

E

Finished in 0.001042s, 959.6929 runs/s, 0.0000 assertions/s.

1) Error:
TestPetShop#test_remove_pet_by_name:
NoMethodError: undefined method `find_pet_by_name' for #<TestPetShop:0x007ff68b6be98>
specs/pet_shop_spec.rb:142:in `test_remove_pet_by_name'
1 runs, 0 assertions, 0 failures, 1 errors, 0 skips
→ start_point git:(master) ✘ ruby specs/pet_shop_spec.rb
Run options: --seed 8236

# Running:

.

Finished in 0.000920s, 1086.9565 runs/s, 1086.9565 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
→ start_point git:(master) ✘ ruby specs/pet_shop_spec.rb
Run options: --seed 31527

# Running:

nil

.

Finished in 0.000953s, 1049.3180 runs/s, 1049.3180 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
→ start_point git:(master) ✘ ruby specs/pet_shop_spec.rb
Run options: --seed 25520

# Running:

.

Finished in 0.001012s, 988.1423 runs/s, 988.1423 assertions/s.

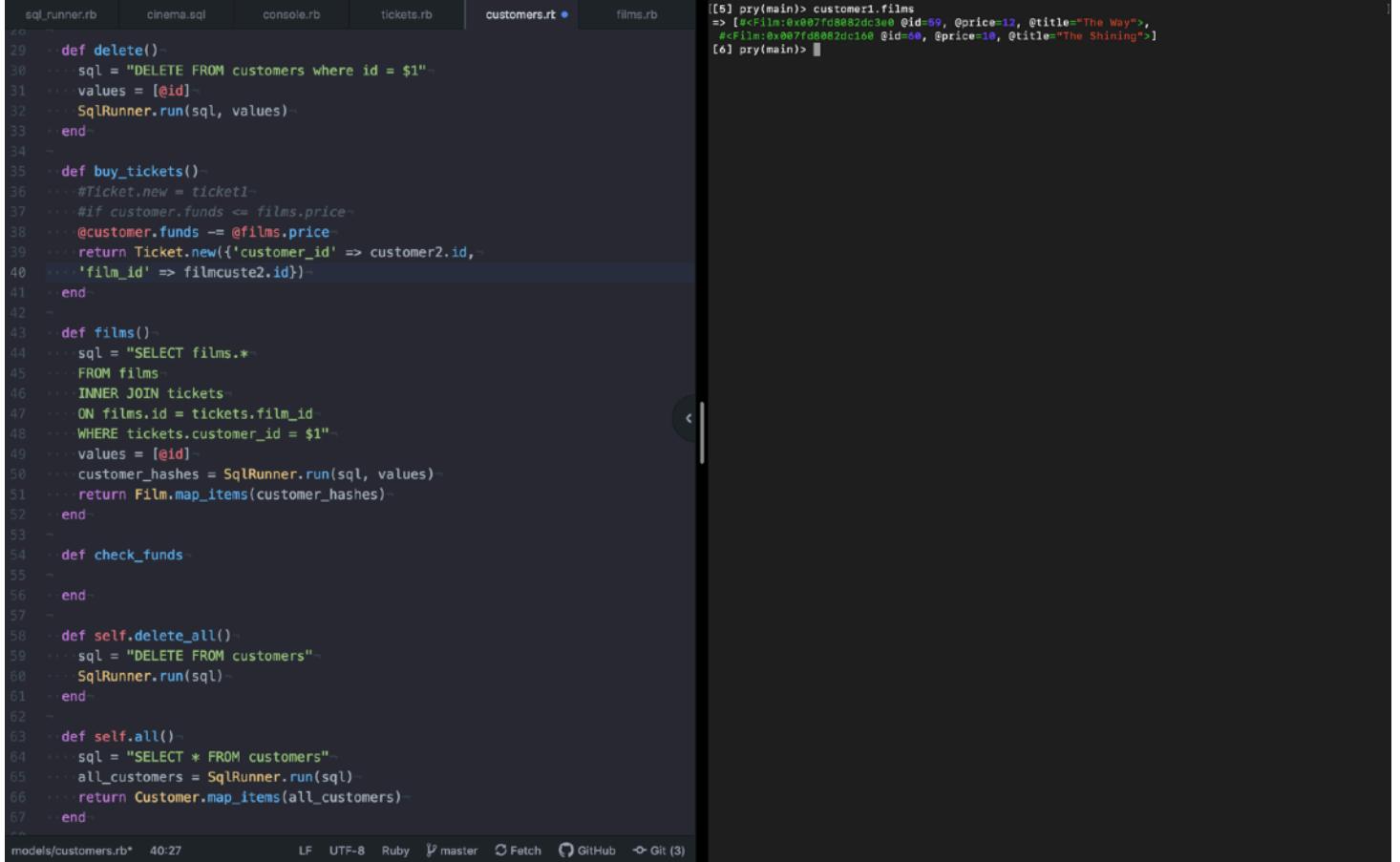
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
→ start_point git:(master) ✘

```

The left-hand screenshot shows an array (an array of hashes) being used for the methods. In the large screen shot you can see a function and its test. This function removes a pet from the array. In the terminal you can see that the test passes (bottom of terminal) and the results have also been printed - Arthur the dog is no longer present in the list.

Week 3

Unit	Ref	Evidence
I&T	I.T.3	Demonstrate searching data in a program. Take screenshots of: *Function that searches data *The result of the function running



```
sql_runner.rb  cinema.sql  console.rb  tickets.rb  customers.rb*  films.rb
[[5] pry(main)> customer1.films
=> [<Film:0x07fc8882dc3e0 @id=50, @price=12, @title="The Way">,
 #<Film:0x07fd8082dc160 @id=60, @price=10, @title="The Shining">]
[6] pry(main)> |
```

```
sql_runner.rb  cinema.sql  console.rb  tickets.rb  customers.rb*  films.rb
[[5] pry(main)> customer1.films
=> [<Film:0x07fc8882dc3e0 @id=50, @price=12, @title="The Way">,
 #<Film:0x07fd8082dc160 @id=60, @price=10, @title="The Shining">]
[6] pry(main)> |
```

```
models/customers.rb*  40:27  LF  UTF-8  Ruby  master  Fetch  GitHub  Git (3)
```

The screenshot shows the method `films()` being used and called. This allows for customer data to be searched and to show which film they are watching/linked to. In the terminal `customer1.films` has been called and it shows they have watched “The Way” and “The Shining”

Unit	Ref	Evidence
I&T	I.T.4	Demonstrate sorting data in a program. Take screenshots of: *Function that sorts data *The result of the function running

```

star_system.rb -- ~/codeclan_work/week_02/day_4/day_4/enumeration_la...
star_system.spec.rb
  @solar_system.get_planet_by_name("Mars")-
38  # assert_equal(@mars, result)-
39  #
40  #
41  # def test_get_largest_planet-
42  #   result =
43  #   @solar_system.get_largest_planet-
44  #   assert_equal(@jupiter, result)-
45  #
46  # def test_get_smallest_planet-
47  #   result =
48  #   @solar_system.get_smallest_planet-
49  #   assert_equal(@pluto, result)-
50  #
51  #
52  # def test_get_planets_with_no_moons-
53  #   expected_planets = [ @mercury, @venus ]
54  #   result =
55  #   @solar_system.get_planets_with_no_moons-
56  #   assert_equal(expected_planets, result)-
57  #
58  # def
59  #   test_get_names_of_planet_with_more_than_fou
60  #   r_moons-
61  #   expected_names = ["Jupiter",
62  #   "Neptune", "Saturn", "Uranus"]-
63  #   result =
64  #   @solar_system.get_planets_with_more_moons(4
65  #   )-
66  #   assert_equal(expected_names, result)-
67  #
68  #
69  # def
70  #   test_number_of_planets_less_than_1billion_k
71  #   m_from_sun-
72  #   result =

```

star_system.rb 13:1

artirst.rb -- ~/codeclan_homeworks/week_03/w3d3

planet.rb star_system.rb

```

1 class StarSystem
2   attr_reader :name, :planets
3   #
4   def initialize(name, planets)
5     @name = name
6     @planets = planets
7   end
8   #
9   def get_smallest_planet()
10    @planets.min{|planet_a, planet_b|
11      planet_a.diameter <= planet_b.diameter}
12  end
13

```

./weekend_3_hm -zshb/start_point -zsh ...

```

+ start_point ruby specs/star_system_spec.rb
Run options: --seed 47051

#<Planet:0x007f8be0c19b8 @name="Pluto", @diameter=2274, @distance_from_sun=5913
, @number_of_moons=4>

Finished in 0.000894s, 1118.5682 runs/s, 1118.5682 assertions/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
⇒ start_point []

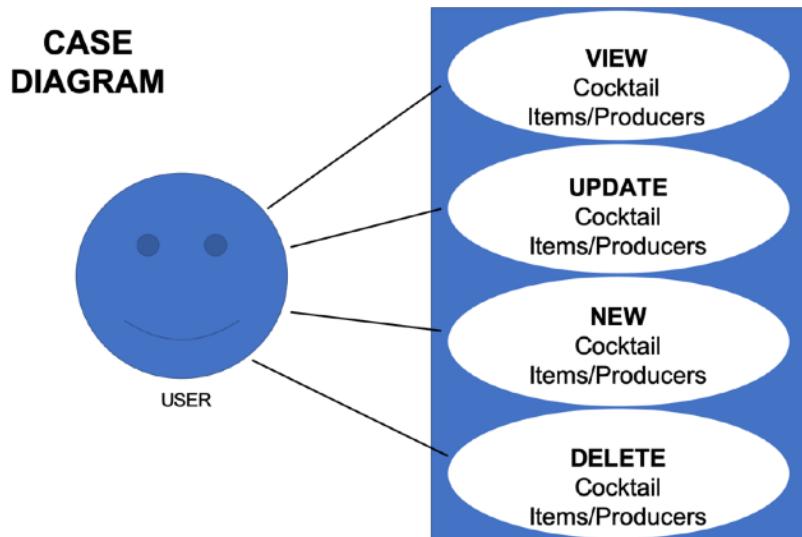
```

LF UTF-8 Ruby GitHub Git (0)

In the above screenshot the function sorts through the data (in this case planets) using .min and returns the smallest planet. An enumerator is used to sort through the data before returning the value. The result of this can be seen in the screenshot.

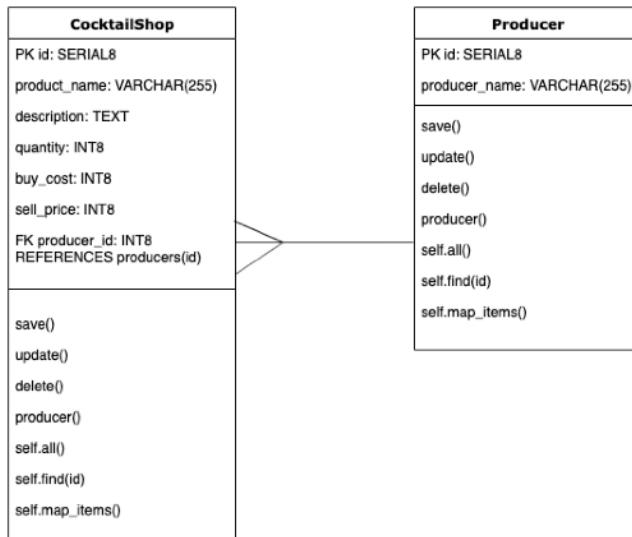
Week 4

Unit	Ref	Evidence
A&D	A.D.1	A Use Case Diagram



This use case diagram give an overview of what is needed from the software being developed. In this instance it can be seen that functionality to view, update, create new and delete items or producers is necessary.

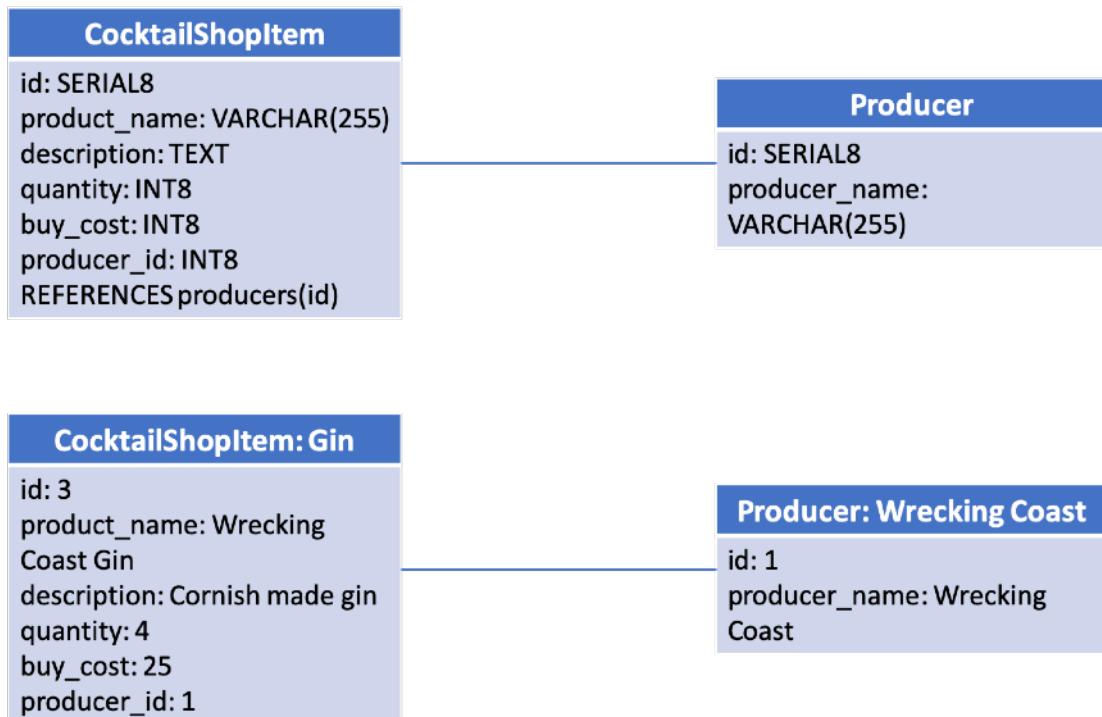
Unit	Ref	Evidence
A&D	A.D.2	A Class Diagram



This class diagram outlines the structure of the system being created and includes the class names, attributes required for each class, the type of each attribute, the methods required and the relationship between the classes.

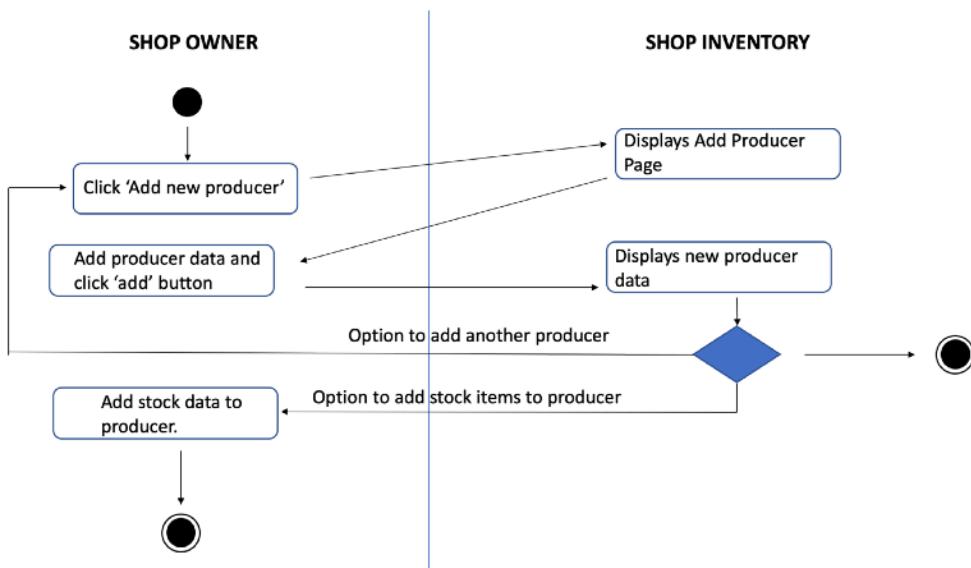
In the above example we have a one (Producer) to many (CocktailShop) relationship. Producer has an attribute of producer_name with the type of VARCHAR(255). Methods required for the class include save(), update() and delete().

Unit	Ref	Evidence
A&D	A.D.3	An Object Diagram



The object diagram, which is similar to the class diagram, shows the classes being created and the attributes required. The object diagram shows how an actual class is set up and used. In this instance the **CocktailShopItem** have seven attributes which have been allocated the relevant data.

Unit	Ref	Evidence
A&D	A.D.4	An Activity Diagram



The activity diagram shows the flow from one step to another. This diagram details the process of adding a new producer to the shop inventory.

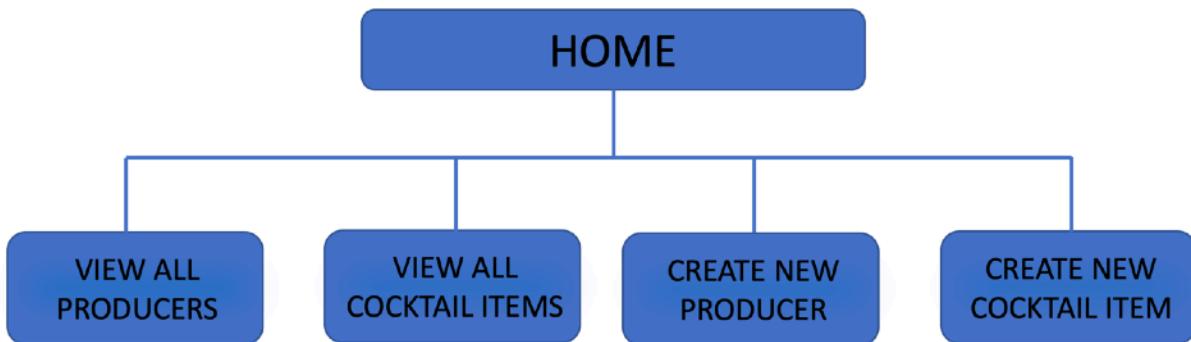
Unit	Ref	Evidence
A&D	A.D.6	<p>Produce an Implementations Constraints plan detailing the following factors:</p> <ul style="list-style-type: none"> *Hardware and software platforms *Performance requirements *Persistent storage and transactions *Usability *Budgets *Time

Constraint Category	Implementation Constraint	Solution
Hardware and Software Platforms	Different shops may have different computer hardware which would also vary the software platforms they are using. This could be an issue if the provided app does not run on older/more limited web browsers or software platforms. For example the app may not work on older versions of internet explorer or firefox.	Design the app to run on a free and widely used web browser such as Google Chrome. In addition provide installation notes in the app's readme file.
Performance Requirements	Computers running the app may not have enough memory to run the app quickly or efficiently. This could be a problem if severe delays are added to the user, such as long load times to open the application or to search for items.	Ensure the app code is written as efficiently as possible and also provide minimum recommend software requirements for the user to ensure successful operation.
Persistent Storage and Transaction	Possible issues include users being unable to access the database due to poor or no internet connectivity. This could result in the app being unusable without a strong internet connection.	Provide minimum internet speed information for users. Create an offline version of the app which can be accessed without internet and will then update once connected.
Usability	Potential issue for users with additional needs, i.e. a visual impairment. This could reduce accessibility for some users. For example, if the user had a visual impairment, they may not be able to use the application with a screen reader if alt tags, etc. haven't been set up correctly.	Follow accessibility guidelines when creating application and ensure to use accessible colour schemes and alt tags for images.
Budgets	It may not be possible to provide all of the users desired functionality with the budget they have set for the project/app. This could result in the client/user receiving less than they had expected/needed. For example the may receive functionality to add and delete new stock but not to edit it.	Break down functionality into order of priority and cost. Agree with client which items are most needed and should be prioritized.
Time Limitations	The client may have a short timeframe in which they desire the project to be completed in. This could result in not all areas of the app being completed on time. Being aware of time limitations will also help to ascertain how many staff are needed to complete the project in the given time.	To create a timeline/work flow for the project to ascertain how long it will take to create and provide options such as an app with reduced functionality or taking on additional staff to ensure the app is completed within the required timeframe.

The above Implementations Constraints plan displays a number of areas where issues/constraints could be experienced during the app creation. Each category item is broken down into what the constraint could be, how it could effect the product and why it is an issue. A solution is provided for each category item.

Unit	Ref	Evidence
P	P.5	User Site Map

The sitemap provides a layout of the app/websites page layout and how they can be accessed, i.e. the ‘view all producers’ page can be accessed from the homepage.



Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams

HOME PAGE

HEADER IMAGE

VIEW ALL SEARCH ADD NEW ITEM

WELCOME/INTRO TEXT

ADD NEW ITEM

HEADER IMAGE

VIEW ALL SEARCH

Product Name:

Product Description:

Quantity:

Purchase Cost: £

Sell Price: £

Producer Name:

ADD PRODUCT

The above wireframe shows the potential design for two pages within the app. The first displays the homepage/initial landing page and the second shows a form for adding new items to the database.

Week 5

Unit	Ref	Evidence
P	P.10	Example of Pseudocode used for a method

```
function sumUp(array) {
  var total = 0;
  for (var item of array) {
    total += item;
  }
  return total;
}
```

The above code returns the sum of the data passed to it. The function is declared as sumUp and is passed in an argument (in this case array).

To start off the total is set to 0, otherwise the code doesn't know what the start point of sum function is.

The next line is known as a for loop. It uses the passed in argument and loops through each item individually. The next line of code tells it to add each item to the total. Every time an item is added it will create a new total before adding the next item.

Finally the function is told to give back the total once the argument has been looped through and all items added to the total.

Unit	Ref	Evidence
P	P.13	Show user input being processed according to design requirements. Take a screenshot of: <ul style="list-style-type: none"> * The user inputting something into your program * The user input being saved or used in some way



[HOME](#)
[VIEW ALL PRODUCERS](#)
[VIEW ALL COCKTAIL ITEMS](#)
[CREATE NEW COCKTAIL ITEM](#)
[CREATE NEW PRODUCER](#)

Add a new Cocktail Item

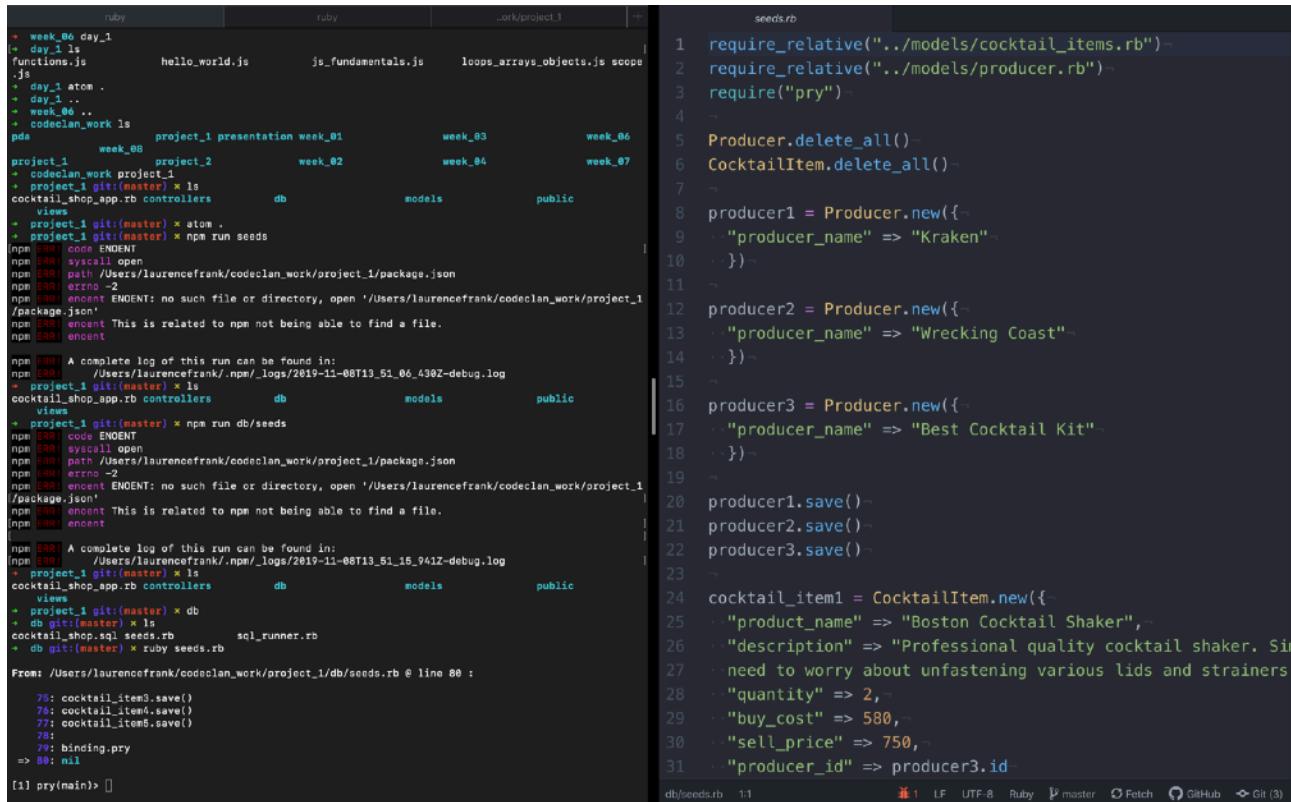
Product Name: Product Description: Quantity: Purchase Cost: Sell Price: Producer Name:

Cocktail Items

Product Name	Description	Quantity	Buy Cost	Sell Price	Mark Up	Producer	Stock Level
Boston Cocktail Shaker	<ul style="list-style-type: none"> • Professional quality cocktail shaker. • Simple to use - no need to worry about unfastening various lids and strainers. 	2	£5.8	£7.5	£1.7	Best Cocktail Kit	LOW STOCK
Jigger	<ul style="list-style-type: none"> • Doubled-ended alcohol measure: One side measures single shots (25 ml) the other doubles (50 ml). • The cocktail jigger is made of durable, rust-resistant, food-safe stainless steel. 	0	£3.9	£5.2	£1.3	Best Cocktail Kit	NO STOCK
Cocktail Strainer	<ul style="list-style-type: none"> • Hawthorn Stainless Steel Strainer. • 4 Prong. • Use with a Boston Cocktail Shaker. 	3	£3.85	£4.9	£1.05	Best Cocktail Kit	LOW STOCK
Wrecking Coast Gin	<ul style="list-style-type: none"> • Clotted cream gin. • 70cl. 	9	£35.0	£40.0	£5.0	Wrecking Coast	
Kraken Rum	<ul style="list-style-type: none"> • Black Spiced Rum. • 70cl. 	9	£18.0	£22.0	£4.0	Kraken	
Measuring Jug	<ul style="list-style-type: none"> • Small measuring jug. 	3	£1.0	£1.5	£0.5	Best Cocktail Kit	LOW STOCK

The first screenshot shows a user adding a new item into the program, in this instance a measuring jug. The second screenshot shows the list of items/products in the program with the measuring jug now added to the list of items.

Unit	Ref	Evidence
P	P.14	Show an interaction with data persistence. Take a screenshot of: <ul style="list-style-type: none"> * Data being inputted into your program * Confirmation of the data being saved



```

ruby          ruby          .fork/project_1
+- week_06 day_1
  + day_1 ls      hello_world.js      js_fundamentals.js    loops_arrays_objects.js scope.js
  + day_1 atom .
  + day_1 ..
  + week_06 ..
  + codeclan_work ls
    week_08   project_1 presentation week_01           week_03      week_06
  project_1       project_2           week_02           week_04      week_07
  + codeclan_work project_1
  + project_1 git:(master) * ls
cocktail_shop_app.rb controllers      db            models      public
  views
+ project_1 git:(master) * atom .
+ project_1 git:(master) * npm run seeds
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /Users/laurencefrank/codeclan_work/project_1/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/Users/laurencefrank/codeclan_work/project_1/package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR!     /Users/laurencefrank/.npm/_logs/2019-11-08T19_51_06_430Z-debug.log
+ project_1 git:(master) * ls
cocktail_shop_app.rb controllers      db            models      public
  views
+ project_1 git:(master) * npm run db/seeds
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /Users/laurencefrank/codeclan_work/project_1/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/Users/laurencefrank/codeclan_work/project_1/package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR!     /Users/laurencefrank/.npm/_logs/2019-11-08T19_51_15_941Z-debug.log
+ project_1 git:(master) * ls
cocktail_shop_app.rb controllers      db            models      public
  views
+ project_1 git:(master) * db
+ db git:(master) * ls
cocktail_shop_app.rb seeds.rb        sql_runner.rb
+ db git:(master) * ruby seeds.rb

From: /Users/laurencefrank/codeclan_work/project_1/db/seeds.rb @ line 80 :

  75: cocktail_item3.save()
  76: cocktail_item4.save()
  77: cocktail_item5.save()
  78:
  79: binding.pr
=> 80: nil

[1]: pry(main)> []

```

```

seeds.rb
1 require_relative("../models/cocktail_items.rb")
2 require_relative("../models/producer.rb")
3 require("pry")
4
5 Producer.delete_all()
6 CocktailItem.delete_all()
7
8 producer1 = Producer.new({
9   "producer_name" => "Kraken"
10 })
11
12 producer2 = Producer.new({
13   "producer_name" => "Wrecking Coast"
14 })
15
16 producer3 = Producer.new({
17   "producer_name" => "Best Cocktail Kit"
18 })
19
20 producer1.save()
21 producer2.save()
22 producer3.save()
23
24 cocktail_item1 = CocktailItem.new({
25   "product_name" => "Boston Cocktail Shaker",
26   "description" => "Professional quality cocktail shaker. Simple need to worry about unfastening various lids and strainers",
27   "quantity" => 2,
28   "buy_cost" => 580,
29   "sell_price" => 750,
30   "producer_id" => producer3.id
31 })

db/seeds.rb 1-1
  ↻ 1 LF  UTF-8 Ruby ⌂ Fetch GitHub ⌂ Git (3)

```

Cocktail Item Producers

Producer Name	Products
Kraken	Kraken Rum
Wrecking Coast	Wrecking Coast Gin
Best Cocktail Kit	Boston Cocktail Shaker Jigger Cocktail Strainer

Screenshot 1 shows data being inputted/saved into the program using a seeds file. Screenshot 2 shows the data now present in the program after being saved.

Unit	Ref	Evidence
P	P.15	Show the correct output of results and feedback to user. Take a screenshot of: * The user requesting information or an action to be performed * The user request being processed correctly and demonstrated in the program

Cocktail Item Producers

Producer Name	Products
Kraken	Kraken Rum
Wrecking Coast	Wrecking Coast Gin
Best Cocktail Kit	Boston Cocktail Shaker Jigger Cocktail Strainer

Producer

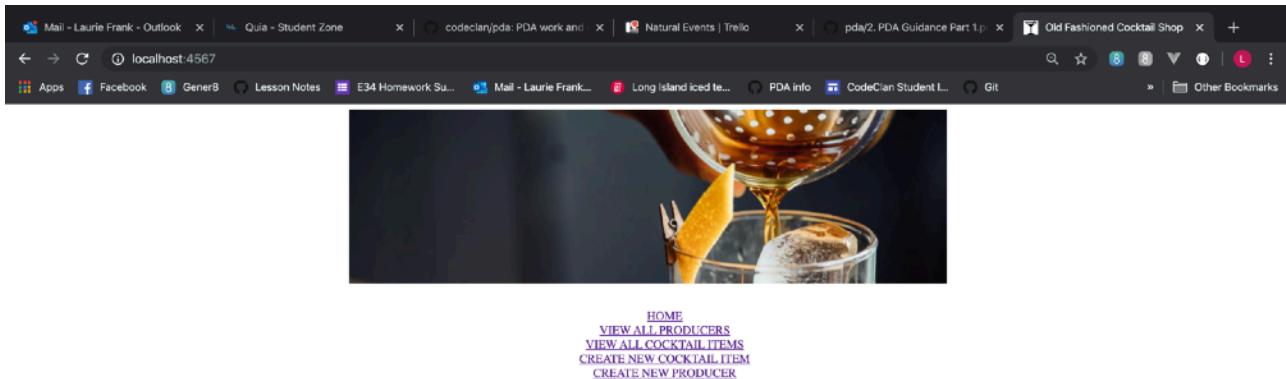
Producer Name	Products
Best Cocktail Kit	32

[Edit Item](#)

[Delete](#)

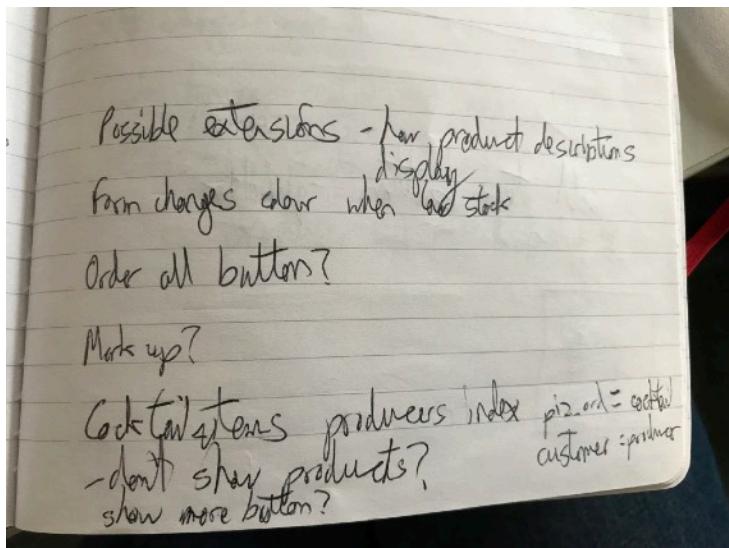
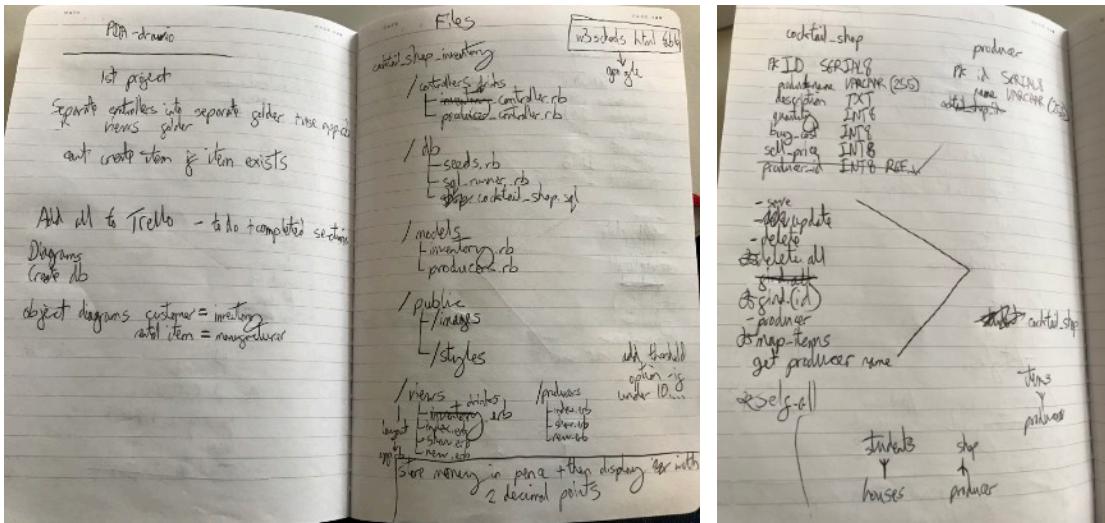
Screenshot 1 shows a list of items, when the item ‘Best Cocktail Kit’ is clicked on the returned request is shown (screenshot 2). In this instance the returned request is the available data which is the number of products the ‘Best Cocktail Kit’ has.

Unit	Ref	Evidence
P	P.11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.



Screenshot of project homepage. Github link: <https://github.com/LaurieFrank/project1>

Unit	Ref	Evidence
P	P.12	Take screenshots or photos of your planning and the different stages of development to show changes.

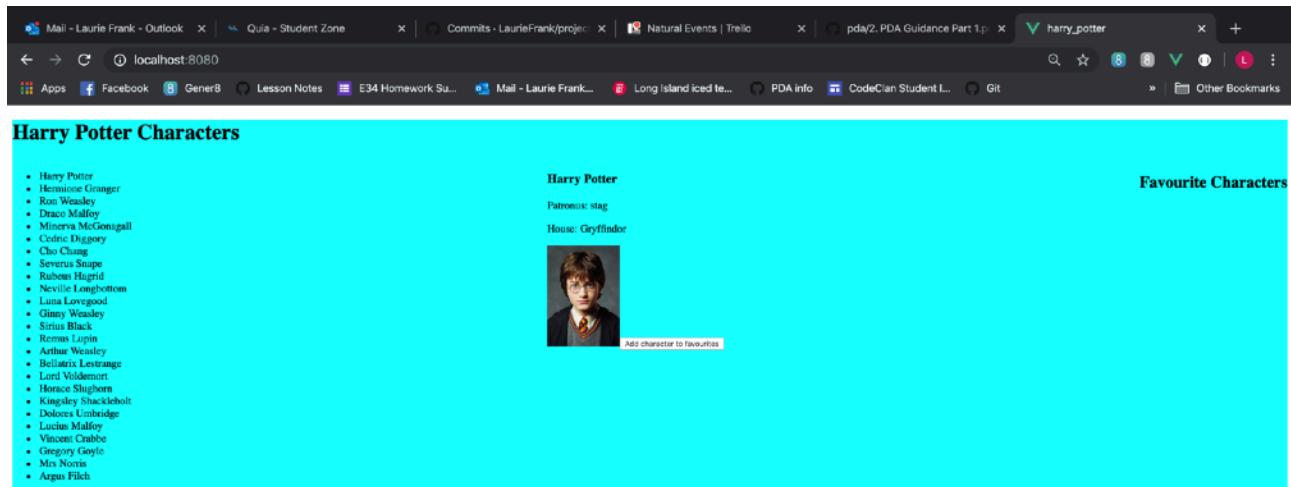


The first two photos show elements of the planning stage (more planning can be seen in the Week 4 diagrams). The third photo shows the projects development once the initial stages had been completed, including items such as an Order All button or displaying product mark up.

Week 7

Unit	Ref	Evidence
P	P.16	Show an API being used within your program. Take a screenshot of: * The code that uses or implements the API * The API being used by the program whilst running

```
mounted(){  
  EventBus.$on("character-selected", (character) => {  
    this.selectedCharacter = character;  
  })  
  
  fetch('http://hp-api.herokuapp.com/api/characters')  
    .then(response => response.json())  
    .then(responseCharacters => this.hpCharacters = responseCharacters)  
    // , then(hpCharacters => console.log(hpCharacters))  
},
```



Screenshot 1 shows some code using an API. Screenshot 2 shows the API being used while the program is running.

Week 8

Unit	Ref	Evidence
P	P.2	Take a screenshot of the project brief from your group project.

Restaurant Booking System:

You have been tasked to create a booking system for a brand new restaurant. The restaurant needs a way to book and arrange tables for customers who are booking over the phone. This system is for the staff to use.

MVP:

Your system must be able to:

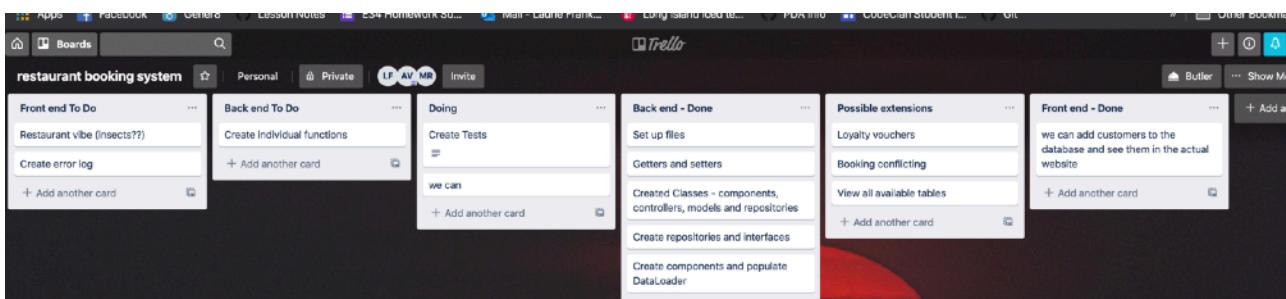
- Allow a customer to book a table at the restaurant for a particular time and date
- Update a booking, for example if the customer wants to change a booking time
- Display a list of bookings for a given date
- Display a list of customers ordered by frequency of visits

Project Extensions:

- Don't allow double bookings
- Add a customer's receipt to a booking so you can view their previous orders and how much they spent
- Calculate how much a customer has spent over a given period of time
- Give discounts to frequent customers
- Whatever features you think would be beneficial to a restaurant

Group project brief - Restaurant Booking System

Unit	Ref	Evidence
P	P.3	Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.



Trello board showing planning stages, work in progress and completed items.

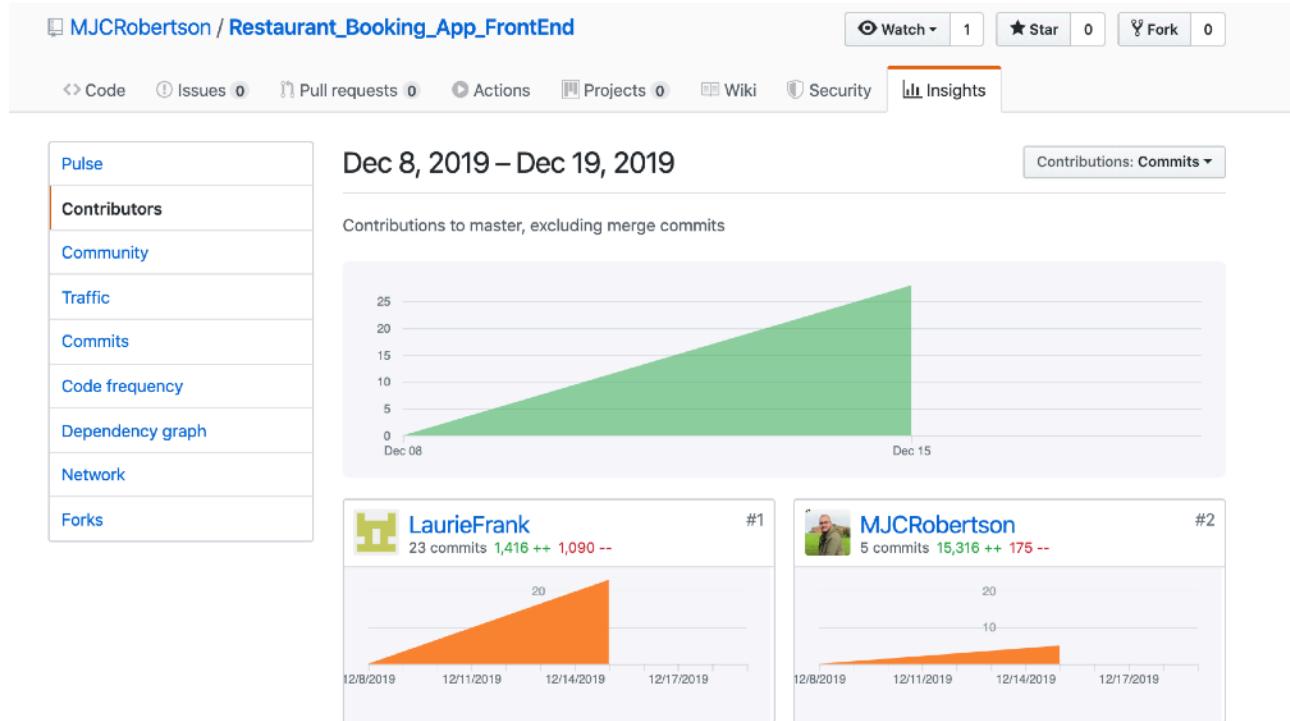
Unit	Ref	Evidence
P	P.4	Write an acceptance criteria and test plan.

Acceptance Criteria	Expected Result	Pass/Fail
A user is able to create a new customer using the 'create new customer' form.	After entering data and pressing submit the data is saved to the API and a full customer list is returned including the new customer.	Pass
A user is able to view all customers.	After clicking on the customers button all customers and their details are shown.	Pass
A user is able to view all tables and if they are available.	After clicking on the tables button details of all tables are shown including whether they are booked or available.	Fail
A user can create a new booking for an existing customer.	A user can click on the create new booking button and select an existing customer from the drop down menu.	Fail
A user can delete a customer.	A user can locate the desired customer to delete from the customer list and after pressing the delete button the customer will be deleted and the user will see a refreshed page without the deleted customer.	Fail

The above plan displays a list of desired criteria, followed by the expected result from the project. Finally the last column confirms with the criteria is met.

Week 9

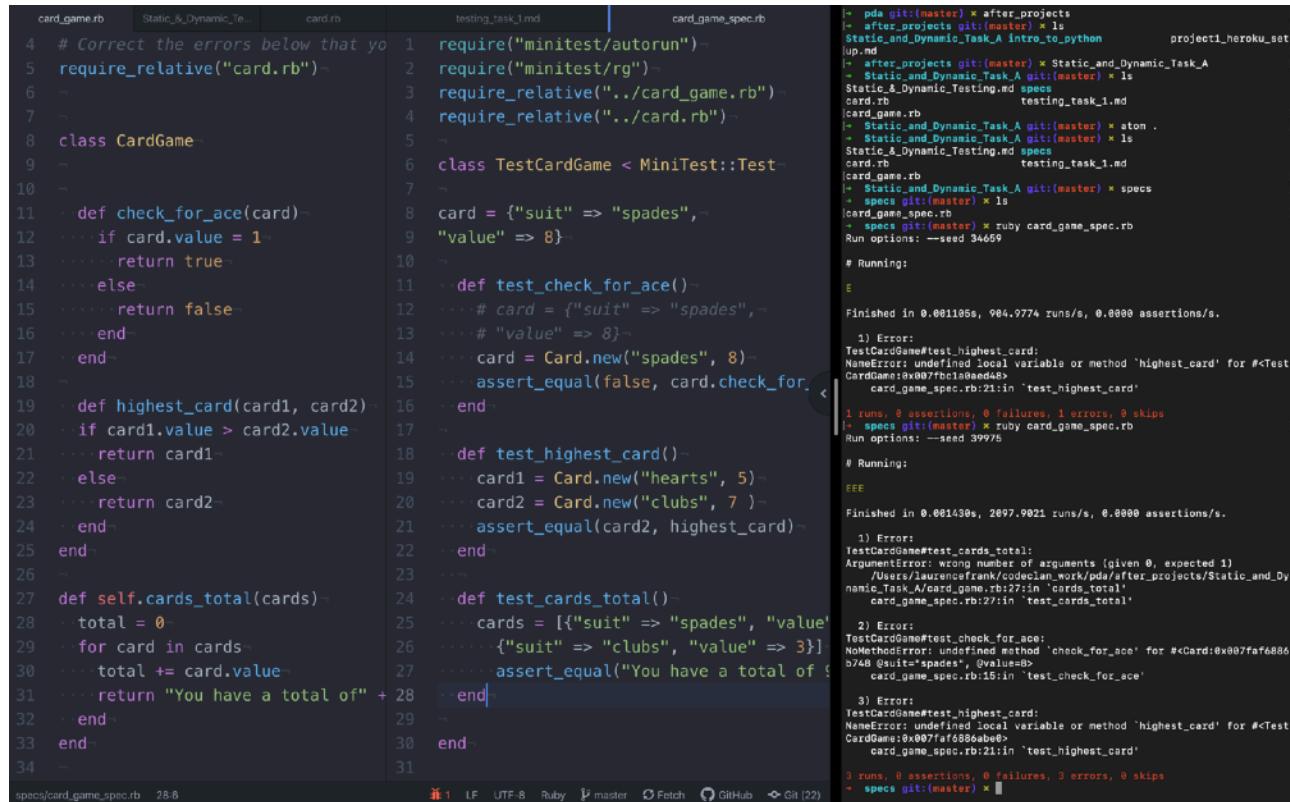
Unit	Ref	Evidence
P	P.1	Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.



The screenshot shows those who have contributed to the project via git push's to the projects repository on GitHub.

Week 11

Unit	Ref	Evidence
P	P.18	Demonstrate testing in your program. Take screenshots of: <ul style="list-style-type: none"> * Example of test code * The test code failing to pass * Example of the test code once errors have been corrected * The test code passing



```

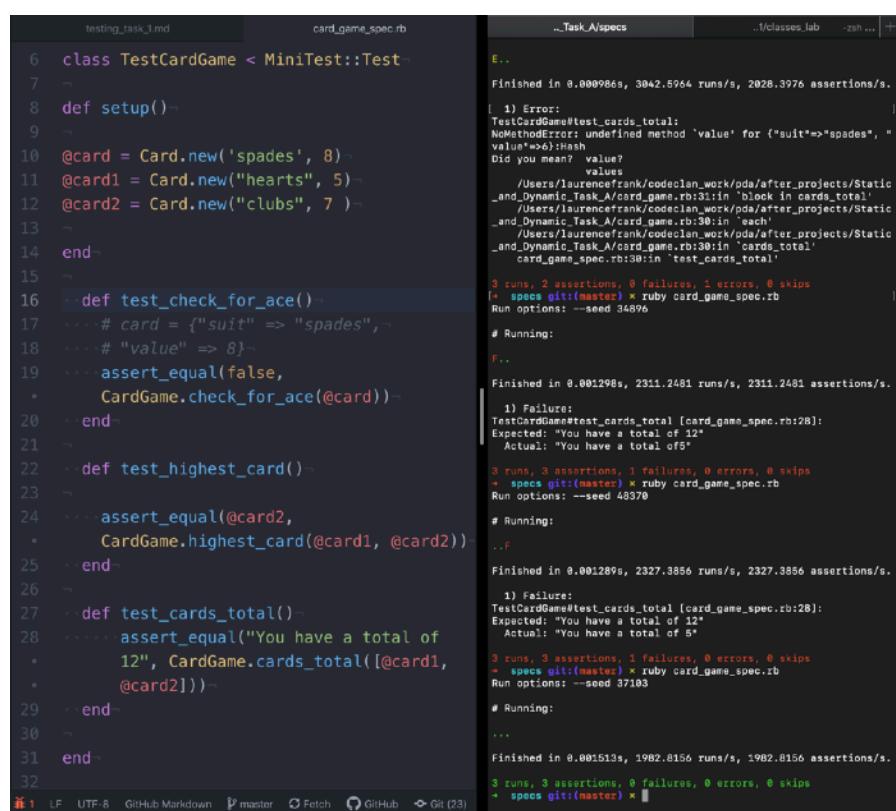
card_game.rb  Static_and_Dynamic_Task_A/intro_to_python/ card.rb  testing_task_1.md  card_game_spec.rb
4 # Correct the errors below that yo 1 require("minitest/autorun")-
5 require_relative("card.rb")- 2 require("minitest/rg")-
6 - 3 require_relative("../card_game.rb")-
7 - 4 require_relative("../card.rb")-
8 class CardGame- 5 -
9 - 10 def check_for_ace(card)- 6 class TestCardGame < MiniTest::Test-
11   if card.value == 1- 7 -
12     return true- 8   card = {"suit" => "spades",-
13   else- 9     "value" => 8}- 10  def test_check_for_ace()- 11    # card = {"suit" => "spades",-
14   end- 12      # "value" => 8}- 13    card = Card.new("spades", 8)- 14    assert_equal(false, card.check_for_ace)- 15  end- 16  def highest_card(card1, card2)- 17    if card1.value > card2.value- 18      return card1- 19    else- 20      return card2- 21    end- 22  def self.cards_total(cards)- 23    total = 0- 24    for card in cards- 25      cards = [{"suit" => "spades", "value" => 1}, {"suit" => "clubs", "value" => 3}]- 26      total += card.value- 27    end- 28    return "You have a total of " + total.to_s- 29  end- 30  end- 31
34
specs/card_game_spec.rb  28:6

```

pdm git:(master) * after_projects
 -> after_projects git:(master) * ls
 Static_and_Dynamic_Task_A intro_to_python project1_heroku_set
 up.md
 -> after_projects git:(master) * Static_and_Dynamic_Task_A
 -> Static_and_Dynamic_Task_A git:(master) * ls
 Static_and_Dynamic_Task_A Testing.md specs
 card.rb testing_task_1.md
 |card_game.rb
 |Static_and_Dynamic_Task_A git:(master) * atom .
 |Static_and_Dynamic_Task_A git:(master) * ls
 Static_and_Dynamic_Task_A Testing.md specs
 card.rb testing_task_1.md
 |Static_and_Dynamic_Task_A git:(master) * specs
 |card_game.rb
 |Static_and_Dynamic_Task_A git:(master) * ls
 |card_game.spec.rb
 |Static_and_Dynamic_Task_A git:(master) * ruby card_game_spec.rb
 Run options: --seed 34659
Running:
E
Finished in 0.001105s, 984.9774 runs/s, 0.0000 assertions/s.
1) Error:
TestCardGame#test_highest_card:
NameError: undefined local variable or method 'highest_card' for #<TestCardGame:0x007fa1c18a6d8>
 card_game_spec.rb:21:in `test_highest_card'
1 runs, 0 assertions, 0 failures, 1 errors, 0 skips
-> specs git:(master) * ruby card_game_spec.rb
Run options: --seed 39978
Running:
EEE
Finished in 0.001430s, 2897.9021 runs/s, 0.0000 assertions/s.
1) Error:
TestCardGame#test_cards_total:
ArgumentError: wrong number of arguments (given 0, expected 1)
/Users/laureneefrank/codeclan/work/pda/after_projects/Static_and_Dynamic_Task_A/card_game_spec.rb:7:in `cards_total'
 card_game_spec.rb:27:in `test_cards_total'
2) Error:
TestCardGame#test_check_for_ace:
NoMethodError: undefined method 'check_for_ace' for #<Card:0x007fa1f6886a0e>
 card_game_spec.rb:15:in `test_check_for_ace'
3) Error:
TestCardGame#test_highest_card:
NameError: undefined local variable or method 'highest_card' for #<TestCardGame:0x007fa1f6886a0e>
 card_game_spec.rb:21:in `test_highest_card'
3 runs, 0 assertions, 0 failures, 3 errors, 0 skips
-> specs git:(master) *

Screenshot 1 shows test code, the code its testing and the terminal window showing the tests failing.

Screenshot 2 shows the edited and fixed test code along with the terminal window showing the tests passing.



```

testing_task_1.md  card_game_spec.rb  ..._Task_A/specs  ..1/classes_lab  -zen ...
6 class TestCardGame < MiniTest::Test- E..
7 -> F.. Finished in 0.000986s, 3042.5964 runs/s, 2028.3976 assertions/s.
8 def setup()-
9 -> F.. 1) Error: TestCardGame#test_cards_total:  

10 @card = Card.new('spades', 8)- NoMethodError: undefined method 'value' for {"suit"=>"spades", "value"=>8}:hash  

11 @card1 = Card.new("hearts", 5)- Did you mean: value?  

12 @card2 = Card.new("clubs", 7 )- values  

13 -> F.. /Users/laureneefrank/codeclan.work/pda/after_projects/Static_and_Dynamic_Task_A/card_game_spec.rb:31:in `block in cards_total'  

14 end- /Users/laureneefrank/codeclan.work/pda/after_projects/Static_and_Dynamic_Task_A/card_game_spec.rb:30:in `each'  

15 -> F.. /Users/laureneefrank/codeclan.work/pda/after_projects/Static_and_Dynamic_Task_A/card_game_spec.rb:30:in `cards_total'  

  card_game_spec.rb:30:in `test_cards_total'  

3 runs, 2 assertions, 0 failures, 1 errors, 0 skips  

-> specs git:(master) * ruby card_game_spec.rb  

Run options: --seed 34896  

# Running:  

F.. Finished in 0.001298s, 2311.2481 runs/s, 2311.2481 assertions/s.  

1) Failure: TestCardGame#test_cards_total [card_game_spec.rb:28]: Expected: "You have a total of 12"  

Actual: "You have a total of 5"  

3 runs, 3 assertions, 1 failures, 0 errors, 0 skips  

-> specs git:(master) * ruby card_game_spec.rb  

Run options: --seed 48378  

# Running:  

..F.. Finished in 0.001289s, 2327.3856 runs/s, 2327.3856 assertions/s.  

1) Failure: TestCardGame#test_cards_total [card_game_spec.rb:28]: Expected: "You have a total of 12"  

Actual: "You have a total of 5"  

3 runs, 3 assertions, 1 failures, 0 errors, 0 skips  

-> specs git:(master) * ruby card_game_spec.rb  

Run options: --seed 37103  

# Running:  

...F.. Finished in 0.001513s, 1982.8156 runs/s, 1982.8156 assertions/s.  

3 runs, 3 assertions, 0 failures, 0 errors, 0 skips  

-> specs git:(master) *
```

Unit	Ref	Evidence
I&T	I.T.1	The use of Encapsulation in a program and what it is doing.

```
public class Flight {

    private ArrayList<Passenger> passengers;
    private PlaneType planeType;
    private String flightNumber;
    private String destination;
    private String departureAirport;
    private String departureTime;

    public Flight(PlaneType planeType,
                  String flightNumber,
                  String destination,
                  String departureAirport,
                  String departureTime){
        this.passengers = new ArrayList<Passenger>();
        this.planeType = planeType;
        this.flightNumber = flightNumber;
        this.destination = destination;
        this.departureAirport = departureAirport;
        this.departureTime = departureTime;
    }
}
```

The above screenshot shows a class ‘Flight’ which is encapsulated. The class’ properties are set to private meaning that they cannot be directly accessed outside of the class itself, unless getters or setters are used.

Week 12

Unit	Ref	Evidence
I&T	I.T.7	The use of Polymorphism in a program and what it is doing.

```
1  public interface ISellable {
2  ...
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9 }
```

```
1  import java.util.ArrayList;
2  ...
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9  ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 }
```

```

1 ~ public abstract class MusicBooks implements ISellable {-
2   ~
3   ·· private String title;-
4   ·· private double buyPrice;-
5   ·· private double sellPrice;-
6   ·· private double quantity;-
7   ~
8 ~ public MusicBooks(String title, double buyPrice, double sellPrice, double quantity){-
9   ··· this.title = title;-
10  ··· this.buyPrice = buyPrice;-
11  ··· this.sellPrice = sellPrice;-
12  ··· this.quantity = quantity;-
13  ··· }-
14 }-
15

```

```

1 public abstract class Instrument implements ISellable {-
2   ~
3   ·· private String manufacturer;-
4   ·· private double buyPrice;-
5   ·· private double sellPrice;-
6   ·· private double quantity;-
7   ~
8   ·· public Instrument(String manufacturer, double buyPrice, double sellPrice, double quantity){-
9     ··· this.manufacturer = manufacturer;-
10    ··· this.buyPrice = buyPrice;-
11    ··· this.sellPrice = sellPrice;-
12    ··· this.quantity = quantity;-
13   ··· }-
14 }-
15

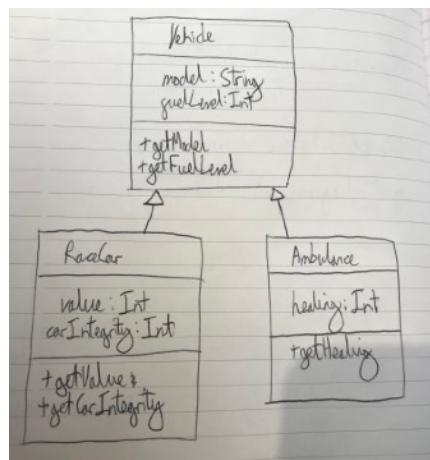
```

Screenshot 1 shows the interface ISellable.

Screenshot 2 shows the ArrayList which can take in different objects, in this instance stock and the method which adds items to the ArrayList.

Screenshots 3 & 4 show the classes which implement ISellable.

Unit	Ref	Evidence
A&D	A.D.5	An Inheritance Diagram



The above image shows an inheritance diagram. The top box shows the overarching class (Vehicle) and the properties are passed down to the RaceCare and Ambulance classes. Both these classes extend from Vehicle and the properties and methods shown are in addition to those passed down by Vehicle.

Unit	Ref	Evidence
I&T	I.T.2	<p>Take a screenshot of the use of Inheritance in a program. Take screenshots of:</p> <ul style="list-style-type: none"> *A Class *A Class that inherits from the previous class *An Object in the inherited class *A Method that uses the information inherited from another class.

```
public abstract class Car implements IDriveable {

    private String model;
    private int fuelLevel;

    public Car(String model, int fuelLevel){
        this.model = model;
        this.fuelLevel = fuelLevel;
    }

    public String drive() { return "vroomy vroomy vroomy"; }

    public void reFuel() { this.fuelLevel = 100; }

    public String getModel() { return this.model; }

    public int getFuelLevel() { return this.fuelLevel; }
}
```

```
public class RaceCar extends Car{

    private int value;
    protected int carIntegrity;

    public RaceCar(String model, int fuelLevel, int value, int carIntegrity){
        super(model, fuelLevel);
        this.value = value;
        this.carIntegrity = carIntegrity;
    }

    public int getValue() { return this.value; }

    public int getCarIntegrity() { return this.carIntegrity; }

    public int hasLostCrash() { return this.carIntegrity -= 1; }
}
```

```
public class RaceCarTest {

    RaceCar raceCar1;
    RaceCar raceCar2;

    @Before
    public void before(){
        raceCar1 = new RaceCar( model: "Rover 75", fuelLevel: 10, value: 111000, carIntegrity: 7 );
        raceCar2 = new RaceCar( model: "Pagani Zonda", fuelLevel: 60, value: 450, carIntegrity: 3 );
    }

    @Test
    public void getCarIntegrity() { assertEquals( expected: 7, raceCar1.getCarIntegrity()); }

}


```

```
@Before
public void before(){
    raceCar1 = new RaceCar( model: "Rover 75", fuelLevel: 10, value: 111000, carIntegrity: 7 );
    raceCar2 = new RaceCar( model: "Pagani Zonda", fuelLevel: 60, value: 450, carIntegrity: 3 );
}

@Test
public void getCarIntegrity() { assertEquals( expected: 7, raceCar1.getCarIntegrity()); }

@Test
public void getRaceCarFuelLevel() { assertEquals( expected: 60, raceCar2.getFuelLevel()); }
```

The above screenshots show the use of inheritance in a program. Screenshot 1 shows a Car class with the properties of model and fuel level. Screenshot 2 shows a RaceCar class which inherits from the Car class. In RaceCar's constructor both model and fuel level are included.

Screenshot 3 shows a RaceCar object in the inherited class and it includes properties from its own class and the inherited class.

Screenshot 4 shows the RaceCar a method inherited from the Car class being used.

Week 14

Unit	Ref	Evidence
P	P.9	Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms.

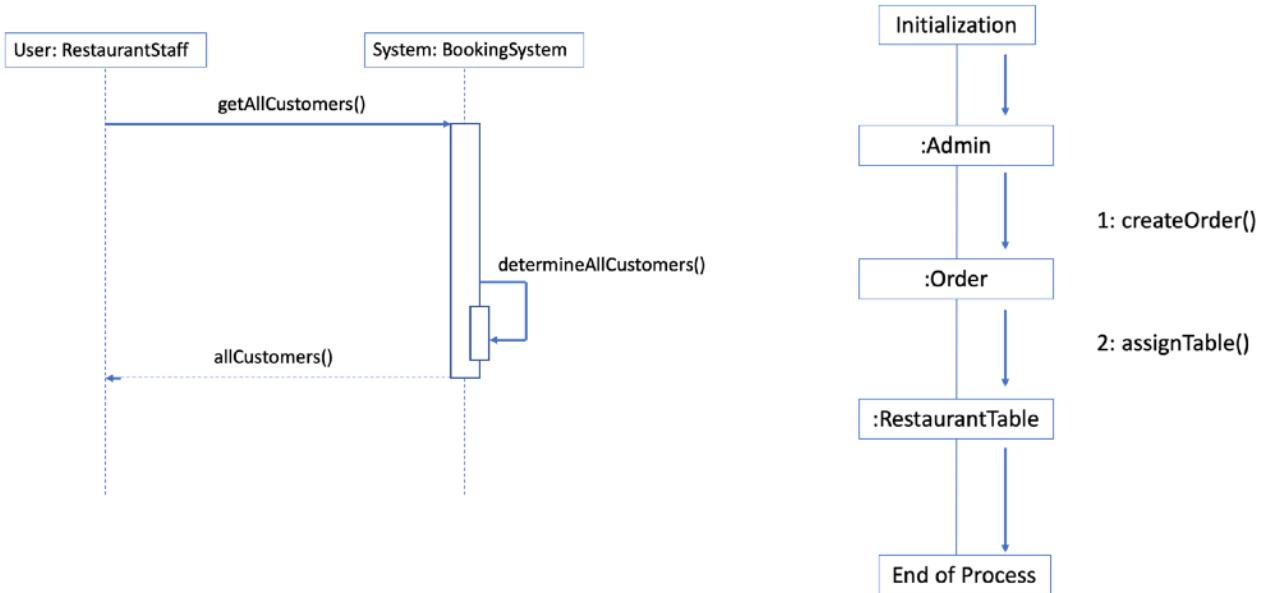
```
public void enlistForRace(RaceCar car){  
    if(car.getValue() <= 500){  
        carsInRace.add(car);  
        vehiclesOnTrack.add(car);  
    }  
}
```

The above screenshot shows an algorithm shown to enter a car into a race in a program. The algorithm was written to ensure that the car has the appropriate properties to be entered into the race. The car is passed in as a variable and the algorithm then determines whether it's value is lower than the set amount, if so the algorithm can add it to the carsInRace and vehiclesOnTrack properties.

```
public void startRace(){  
    if(carsInRace.size() == vehiclesOnTrack.size()){  
        this.raceInProgress = true;  
    }  
}
```

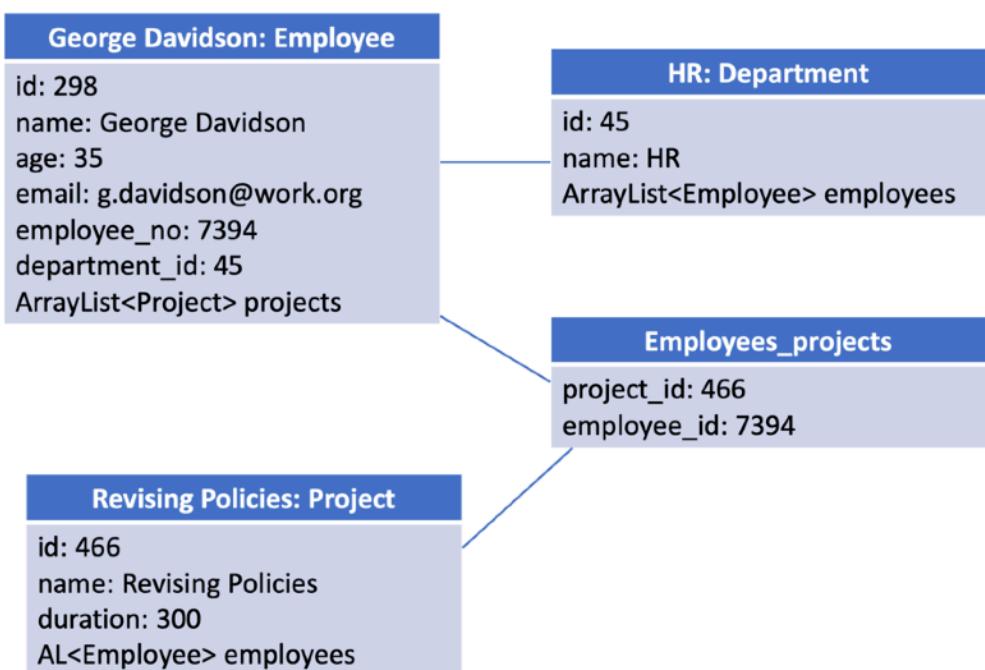
The screenshot above shows an algorithm used to start a race. It was written to make sure that the number of cars in a race are equal to the number of cars on the track. If this is correct, it updates the raceInProgress boolean to true.

Unit	Ref	Evidence
P	P.7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).



The 1st diagram shows a sequence diagram and the 2nd is a collaboration diagram. They both show the order in which messages occur and what messages are sent between the objects in a system.

Unit	Ref	Evidence
P	P.8	Produce two object diagrams.





Two object diagrams are shown above.

Unit	Ref	Evidence
P	P.17	Produce a bug tracking report

Bug/Error	Solution	Date
View all customers link does not do anything	Correct href link	17/12/19
After adding new customers the page doesn't redirect to 'view all customers' page	Ensure redirect path is correct	18/12/19
Component does not render on main app page	Export the component and to main app import list	16/12/19
Edit form does not contain existing customer data	Add default value to each form field using props	18/12/19
Forms do not post to database	Add CORS configuration in Spring to allow the back end of the application to speak to the front end	18/12/19

The above bug tracking reports lists a number of bugs discovered in the application, how they were solved and the date it took place.