Contents

Testing

Objective 9: Markdown

Throughout the construction of the project I added test cards to a GitHub project in my test repository.

These cards test each of the different markdown features which should be included.

Objective 9.ii.a: Code blocks

Short code blocks

Objective 9.ii.a.1 is to display short code blocks within the text body, including code in line with other text.

A short code block is any code block with fewer than 10 lines.

The test card for short code blocks contains the following items which should be displayed as short code blocks:

- A short code block at the start of the text
- A short code block in the middle of a line, between other text
- A short code block within a list item
- A short code block with ten lines
- A short code block at the end of the text

The card used to test this is

~~~
```
System.out.println("Line");
System.out.println("Line 2");
System.out.println("Line 3");
```

Some text with a ```print("Code block")``` in the middle.

A list:

- Item 1
~~~

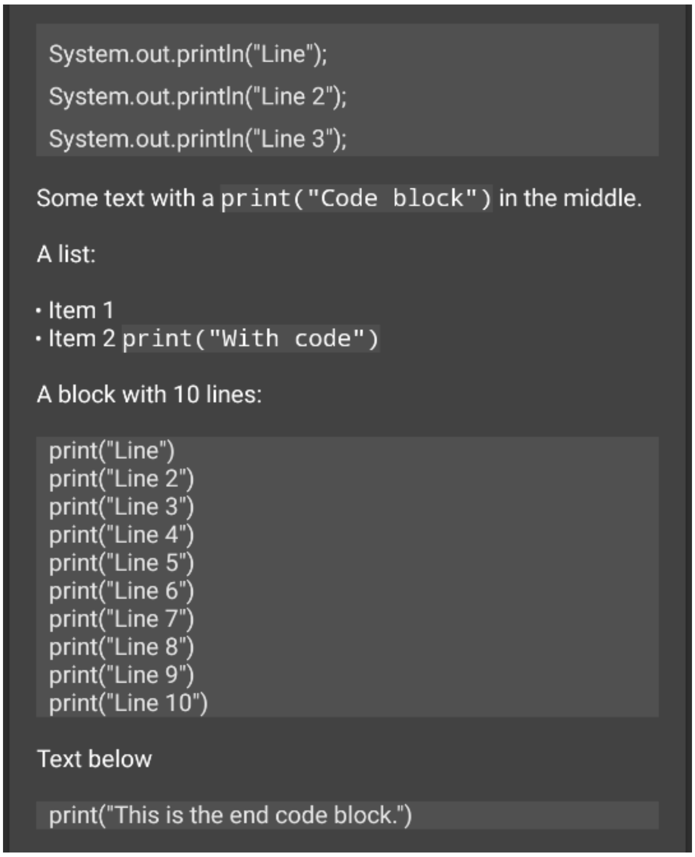
~~~~
- Item 2 ```print(“With code”)```

A block with 10 lines:
```
print(“Line”)
print(“Line 2”)
print(“Line 3”)
print(“Line 4”)
print(“Line 5”)
print(“Line 6”)
print(“Line 7”)
print(“Line 8”)
print(“Line 9”)
print(“Line 10”)
```

Text below
```
print(“This is the end code block.”)
```
~~~~

Test	**Expected Result**	**Result**
Block at start of text	Block shown with grey background	Pass
Block in the middle of a line	Only the part of the line is shown with a grey background	Pass
Block in list item	Only the code part of the list item is shown with a grey background	Pass
Block with ten lines	Block shown with all ten lines within the text	Pass
Block at end of text	Block shown with grey background	Pass

The card is displayed as shown below:



```
System.out.println("Line");
System.out.println("Line 2");
System.out.println("Line 3");
```

Some text with a `print("Code block")` in the middle.

A list:

- Item 1
- Item 2 `print("With code")`

A block with 10 lines:

```
print("Line")
print("Line 2")
print("Line 3")
print("Line 4")
print("Line 5")
print("Line 6")
print("Line 7")
print("Line 8")
print("Line 9")
print("Line 10")
```

Text below

```
print("This is the end code block.")
```

Larger code blocks

Objective 9.ii.a.2 is to display large code blocks as placeholders within the text body, allowing them to be clicked to show the full code block.

A large code block is any code block with more than 10 lines.

The test card for large code blocks contains the following items which should be displayed as large code blocks:

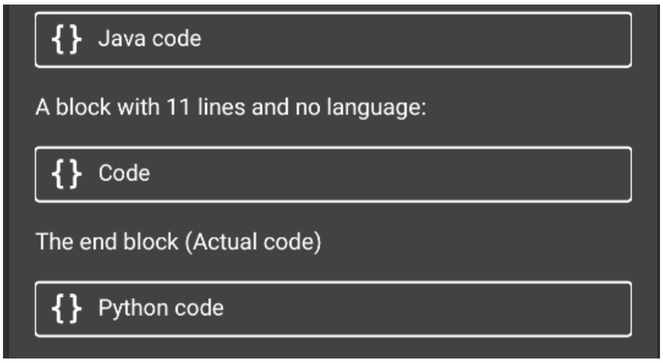
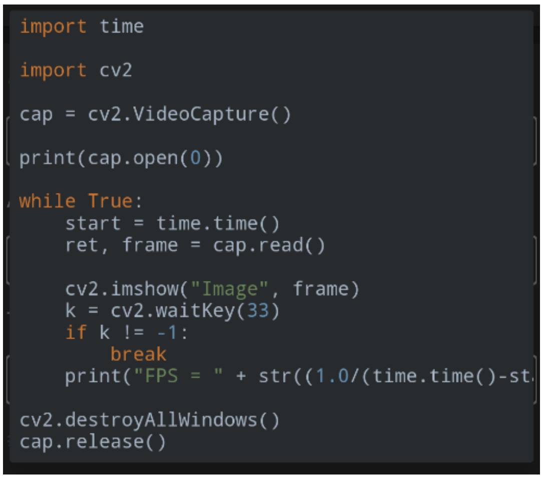
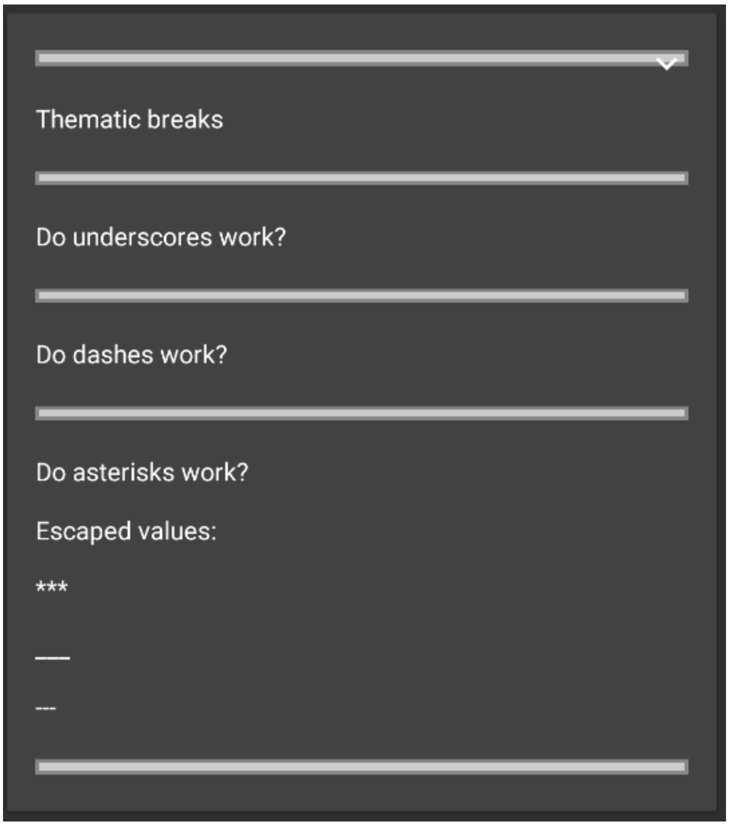
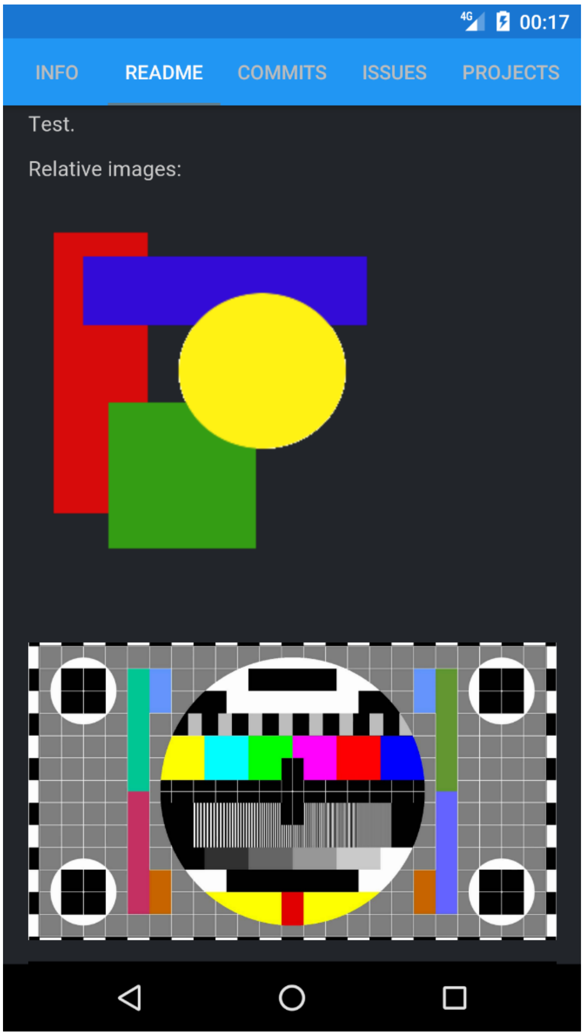
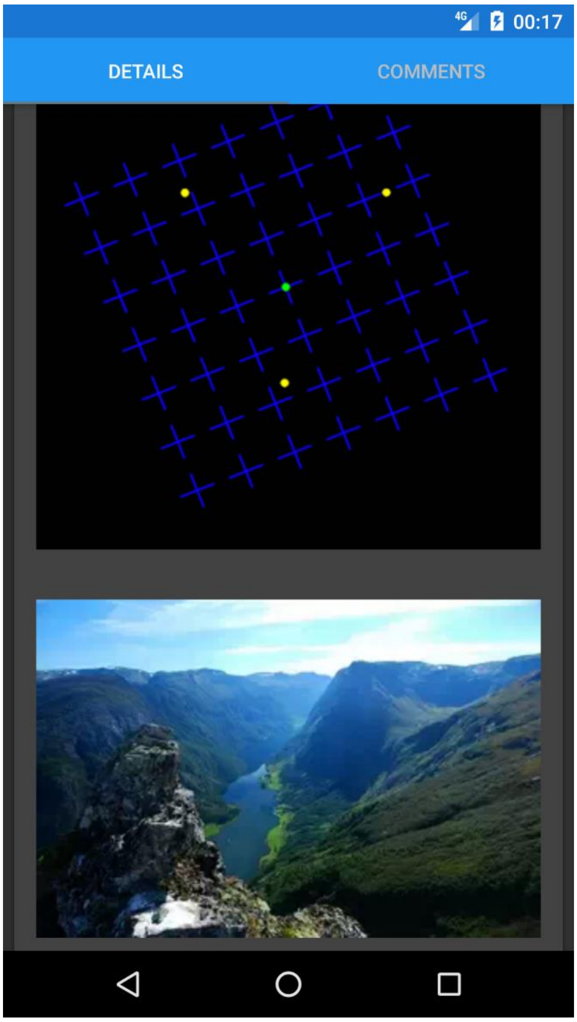
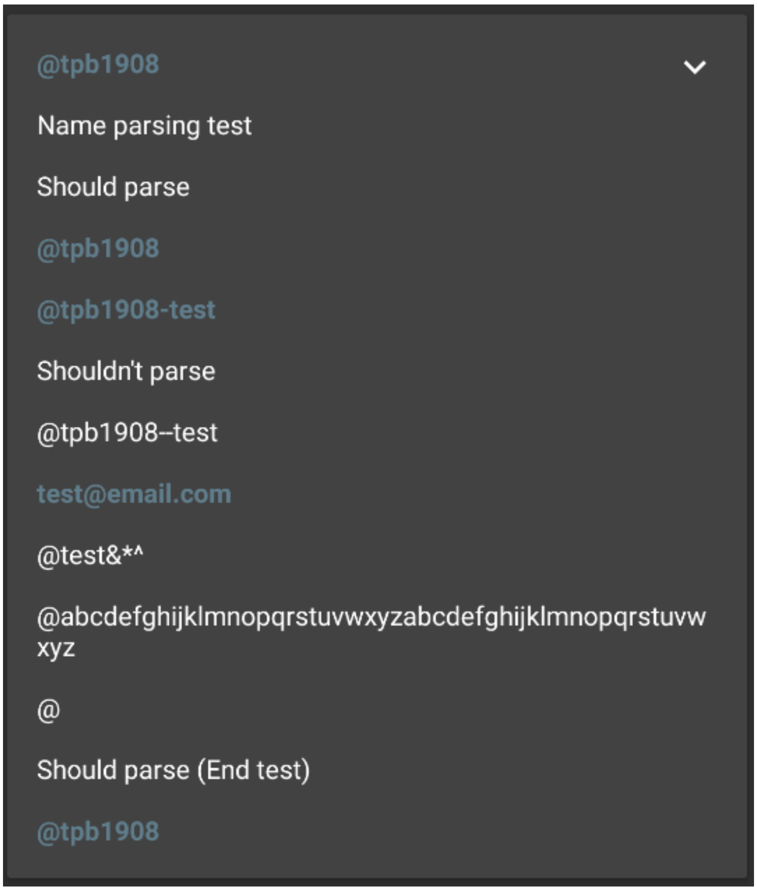
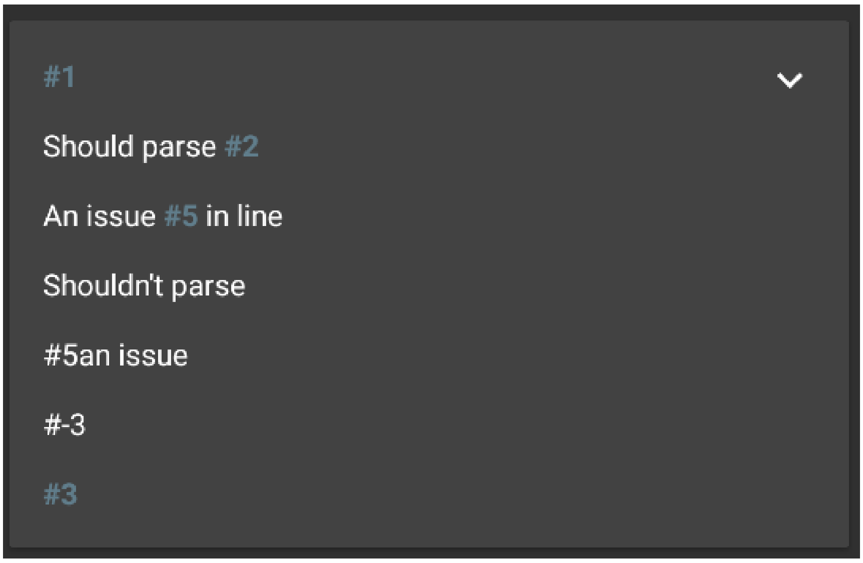
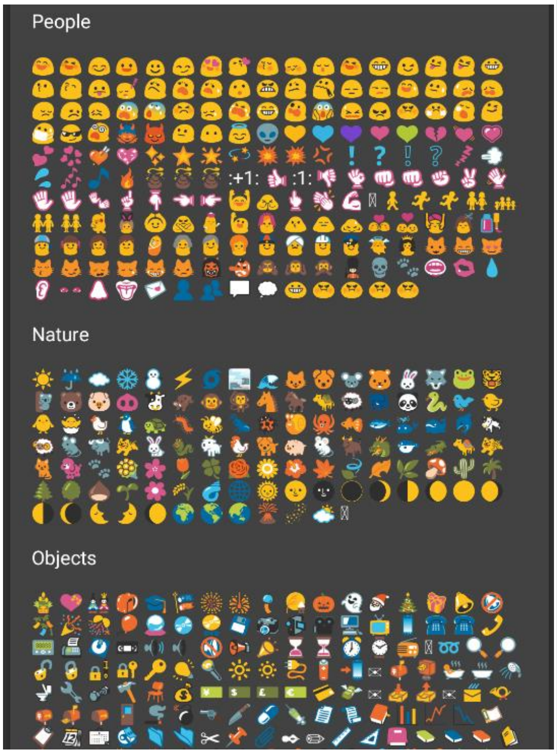
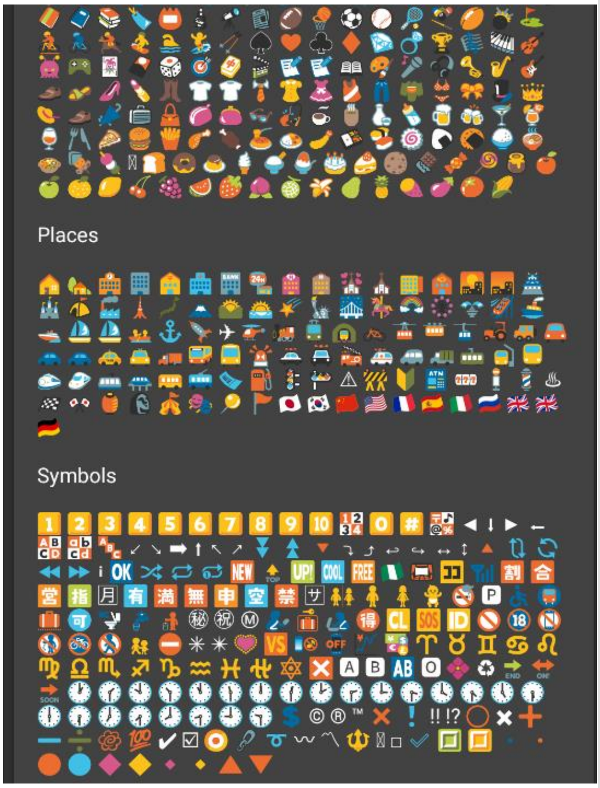
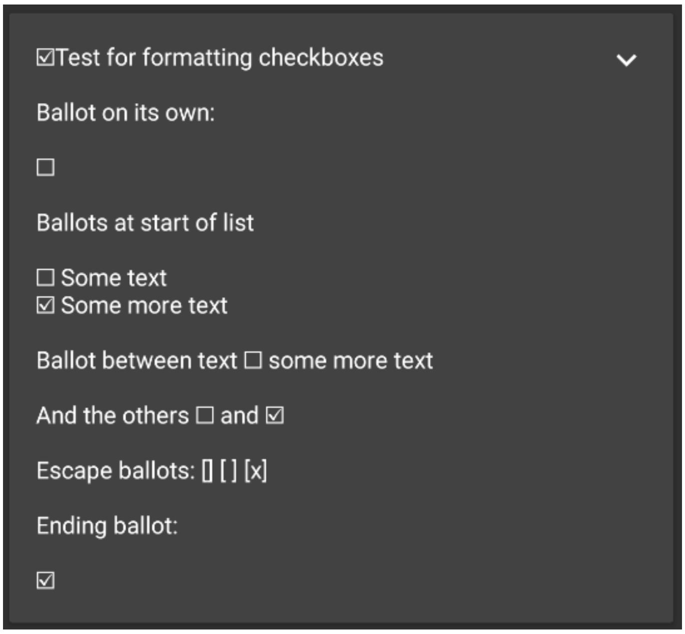
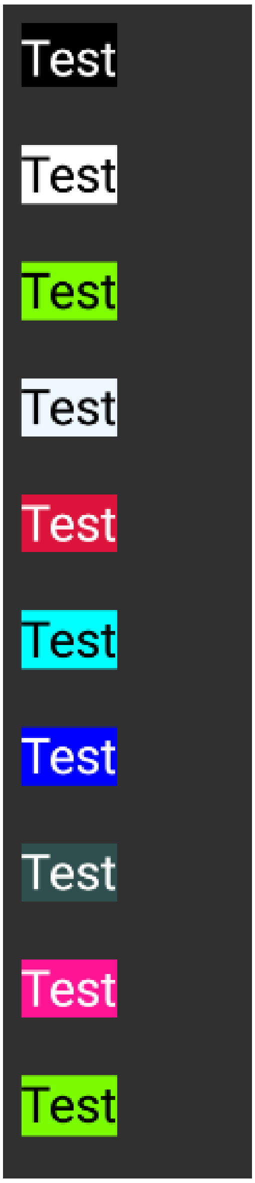
- A large code block at the start of the text
- A large code block with a language
- A large code block without a language
- A large code block with eleven lines
- A large code block at the end of the text

The card text used to test this is:

~~~
``` java
System.out.println(“Line”);
System.out.println(“Line 2”);
System.out.println(“Line 3”);
System.out.println(“Line 4”);
~~~

~~~~
System.out.println(“Line 5”);
System.out.println(“Line 6”);
System.out.println(“Line 7”);
System.out.println(“Line 8”);
System.out.println(“Line 9”);
System.out.println(“Line 10”);
System.out.println(“Line 11”);
System.out.println(“Line 12”);
```

A block with 11 lines and no language:
```
print(“Line”)
print(“Line 2”)
print(“Line 3”)
print(“Line 4”)
print(“Line 5”)
print(“Line 6”)
print(“Line 7”)
print(“Line 8”)
print(“Line 9”)
print(“Line 10”)
print(“Line 11”)
```

The end block (Actual code)
```Python
import time

import cv2

cap = cv2.VideoCapture()

print(cap.open(0))

while True:
start = time.time()
ret, frame = cap.read()

cv2.imshow("Image", frame)
k = cv2.waitKey(33)
if k != -1:
    break
print("FPS = " + str((1.0/(time.time()-start))))
~~~~

~~~
cv2.destroyAllWindows()
cap.release()
```
~~~

Test	Expected Result	Result
Block at start of text	Block placeholder shown	Pass
Block with a language	Block placeholder shown with corresponding language	Pass
Block without language	Block shown without language	Pass
Block clicked	Code dialog shown	Pass
Block at end of text	Block placeholder shown	Pass
Icon displayed before text	Icon was displayed	Pass

The code blocks and code dialog are shown as below:

Code blocks	Code dialog
 {} Java code A block with 11 lines and no language: {} Code The end block (Actual code) {} Python code	

```
import time

import cv2

cap = cv2.VideoCapture()

print(cap.open(0))

while True:
    start = time.time()
    ret, frame = cap.read()

    cv2.imshow("Image", frame)
    k = cv2.waitKey(33)
    if k != -1:
        break
    print("FPS = " + str((1.0/(time.time()-st

cv2.destroyAllWindows()
cap.release()
```

Objective 9.ii.b: Horizontal rules

Horizontal rules should be displayed in the place of any of the three character combinations “—”, “***”, or “___”.

The test card for horizontal rules tests the following which should display a horizontal rule:

- Any of the valid combinations on its own line in the text
- Any of the valid combinations at the start of the text
- Any of the valid combinations at the end of the text

And the following which should not display a horizontal rule:

- Any of the valid combinations in the middle of a line of text
- Any of the valid combinations preceded by a backslash

The card text used to test this is:

—

Thematic breaks

___

Do underscores work?

—

Do dashes work?

***

Do asterisks work?

Escaped values:

***

___

\—

—

Test	Expected Result	Result
Valid combination at start of text	Horizontal rule displayed	Pass
Valid combination on its own line	Horizontal rule displayed	Pass
Valid combination at the end of text	Horizontal rule displayed	Pass
Escaped combinations	Horizontal rule not displayed	Pass

The card is displayed as shown below:



Thematic breaks

Do underscores work?

Do dashes work?

Do asterisks work?

Escaped values:

***

___

Objective 9.ii.c: Image handling and relative link handling

Images with a relative path in the repository should be shown.

In order to test this I added images of each type (png, jpg, gif, bmp, and webp) to the test repository and added relative links to the README as well as a relative path.

```
![Test 1](./test_1.png)
```

```
![Test 2](/test_2.jpg)

![Test 3](test_3.bmp)

![Test_4](./test_4.gif)

![Test 5](/test_5.webp)
```

The links are either the file name, a single forward slash, or a dot slash, which are all valid relative path formats.

The sets of images displayed correctly in both the `MarkdownWebView` and `MarkdownTextView`.
Next, I moved the images to a directory within the repository and updated the links.

```
![Test 1](./test/test_1.png)

![Test 2](/test/test_2.jpg)

![Test 3](test/test_3.bmp)

![Test_4](./test/test_4.gif)

![Test 5](/test/test_5.webp)
```

The images were still displayed in the same manner as before.

MarkdownWebView	`MarkdownTextView`
 Test. Relative images:	

Objective 9.ii.d: Username linking

The test card for username linking tests the following items which should be parsed as username links:

- Usernames linked at the start of the text
- Usernames with a single hyphen
- Usernames at the end of the text

And the following which should not:

- Usernames with two hyphens
- An email

- Usernames longer than 38 characters
- Usernames with invalid characters
- A single "@" character (0 length username)

Test	**Expected Result**	**Result**
Username at the start of the text	Converted to link	Pass
Username with a hyphen	Converted to link	Pass
Username at the end of the text	Converted to link	Pass
Username with two hyphens	Not converted to link	Pass
Email address	Converted to email link	Pass
Username longer than 38 characters	Not converted to link	Pass
Username with invalid characters	Not converted to link	Pass
Single "@" character	Not converted to link	Pass

The display within the app is shown below:



@tpb1908

Name parsing test

Should parse

@tpb1908

@tpb1908-test

Shouldn't parse

@tpb1908--test

test@email.com

@test&*^

@abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvw
xyz

@

Should parse (End test)

@tpb1908

Objective 9.ii.e: Issue linking

The test card for issue linking tests the following items which should be parsed as issue links:

- An issue at the start of the text
- An issue on its own line
- An issue between two other blocks of text
- An issue at the end of the text

And the following which should not:

- An issue which is conjoined with a word
- An issue with a negative number

Test	Expected result	Result
Issue at the start of the text	Converted to link	Pass

Test	Expected result	Result
Issue on its own line	Converted to link	Pass
Issue between two other blocks of text	Converted to link	Pass
Issue at the end of the text	Converted to link	Pass
Issue which is conjoined with a word	Not converted to a link	Pass
Issue with a negative number	Not converted to a link	Pass

The display within the app is shown below:



#1

Should parse #2

An issue #5 in line

Shouldn't parse

#5an issue

#-3

#3

Objective 9.ii.f: Emoji parsing

The test card for emoji parsing tests a list of emoji aliases which should be converted to their respective emoji characters.

The raw text is below:

```
People

 :smile:   :laughing:
:blush:   :smiley:   :relaxed:
:smirk:   :heart_eyes:   :kissing_heart:
:kissing_closed_eyes:   :flushed:   :relieved:
:satisfied:   :grin:   :wink:
:stuck_out_tongue_winking_eye:   :stuck_out_tongue_closed_eyes:   :grinning:
:kissing:   :kissing_smiling_eyes:   :stuck_out_tongue:
```

```
:sleeping:   :worried:   :frowning:
:anguished:   :open_mouth:   :grimacing:
:confused:   :hushed:   :expressionless:
:unamused:   :sweat_smile:   :sweat:
:disappointed_relieved:   :weary:   :pensive:
:disappointed:   :confounded:   :fearful:
:cold_sweat:   :persevere:   :cry:
:sob:   :joy:   :astonished:
:scream:  :tired_face:
:angry:   :rage:   :triumph:
:sleepy:   :yum:   :mask:
:sunglasses:   :dizzy_face:   :imp:
:smiling_imp:   :neutral_face:   :no_mouth:
:innocent:   :alien:   :yellow_heart:
:blue_heart:   :purple_heart:   :heart:
:green_heart:   :broken_heart:   :heartbeat:
:heartpulse:   :two_hearts:   :revolving_hearts:
:cupid:   :sparkling_heart:   :sparkles:
:star:   :star2:   :dizzy:
:boom:   :collision:   :anger:
:exclamation:   :question:   :grey_exclamation:
:grey_question:   :zzz:   :dash:
:sweat_drops:   :notes:   :musical_note:
:fire:   :hankey:   :poop:
:shit:   :+1:   :thumbsup:
:1:   :thumbsdown:   :ok_hand:
:punch:   :facepunch:   :fist:
:v:   :wave:   :hand:
:raised_hand:   :open_hands:   :point_up:
:point_down:   :point_left:   :point_right:
:raised_hands:   :pray:   :point_up_2:
:clap:   :muscle:   :metal:   :walking:   :runner:
:running:   :couple:   :family:
:two_men_holding_hands:   :two_women_holding_hands:   :dancer:
:dancers:   :ok_woman:   :no_good:
:information_desk_person:   :raising_hand:   :bride_with_veil:
:person_with_pouting_face:   :person_frowning:   :bow:
:couplekiss:   :couple_with_heart:   :massage:
:haircut:   :nail_care:   :boy:
:girl:   :woman:   :man:
:baby:   :older_woman:   :older_man:
:person_with_blond_hair:   :man_with_gua_pi_mao:   :man_with_turban:
:construction_worker:   :cop:   :angel:
:princess:   :smiley_cat:   :smile_cat:
:heart_eyes_cat:   :kissing_cat:   :smirk_cat:
:scream_cat:   :crying_cat_face:   :joy_cat:
:pouting_cat:   :japanese_ogre:   :japanese_goblin:
:see_no_evil:   :hear_no_evil:   :speak_no_evil:
:guardsman:   :skull:   :feet:
:lips:   :kiss:   :droplet:
:ear:   :eyes:   :nose:
:tongue:   :love_letter:   :bust_in_silhouette:
:busts_in_silhouette:   :speech_balloon:   :thought_balloon:
:feelsgood:   :rage1:
:rage2:   :rage3:   :rage4:

Nature

:sunny:   :umbrella:   :cloud:
```

```
:snowflake:   :snowman:   :zap:
:cyclone:   :foggy:   :ocean:
:cat:   :dog:   :mouse:
:hamster:   :rabbit:   :wolf:
:frog:   :tiger:   :koala:
:bear:   :pig:   :pig_nose:
:cow:   :boar:   :monkey_face:
:monkey:   :horse:   :racehorse:
:camel:   :sheep:   :elephant:
:panda_face:   :snake:   :bird:
:baby_chick:   :hatched_chick:   :hatching_chick:
:chicken:   :penguin:   :turtle:
:bug:   :honeybee:   :ant:
:beetle:   :snail:   :octopus:
:tropical_fish:   :fish:   :whale:
:whale2:   :dolphin:   :cow2:
:ram:   :rat:   :water_buffalo:
:tiger2:   :rabbit2:   :dragon:
:goat:   :rooster:   :dog2:
:pig2:   :mouse2:   :ox:
:dragon_face:   :blowfish:   :crocodile:
:dromedary_camel:   :leopard:   :cat2:
:poodle:   :paw_prints:   :bouquet:
:cherry_blossom:   :tulip:   :four_leaf_clover:
:rose:   :sunflower:   :hibiscus:
:maple_leaf:   :leaves:   :fallen_leaf:
:herb:   :mushroom:   :cactus:
:palm_tree:   :evergreen_tree:   :deciduous_tree:
:chestnut:   :seedling:   :blossom:
:ear_of_rice:   :shell:   :globe_with_meridians:
:sun_with_face:   :full_moon_with_face:   :new_moon_with_face:
:new_moon:   :waxing_crescent_moon:   :first_quarter_moon:
:waxing_gibbous_moon:   :full_moon:   :waning_gibbous_moon:
:last_quarter_moon:   :waning_crescent_moon:   :last_quarter_moon_with_face:
:first_quarter_moon_with_face:   :moon:   :earth_africa:
:earth_americas:   :earth_asia:   :volcano:
:milky_way:   :partly_sunny: :squirrel:

Objects

:bamboo:   :gift_heart:   :dolls:
:school_satchel:   :mortar_board:   :flags:
:fireworks:   :sparkler:   :wind_chime:
:rice_scene:   :jack_o_lantern:   :ghost:
:santa:   :christmas_tree:   :gift:
:bell:   :no_bell:   :tanabata_tree:
:tada:   :confetti_ball:   :balloon:
:crystal_ball:   :cd:   :dvd:
:floppy_disk:   :camera:   :video_camera:
:movie_camera:   :computer:   :tv:
:iphone:   :phone:   :telephone:
:telephone_receiver:   :pager:   :fax:
:minidisc:   :vhs:   :sound:
:speaker:   :mute:   :loudspeaker:
:mega:   :hourglass:   :hourglass_flowing_sand:
:alarm_clock:   :watch:   :radio:
:satellite:   :loop:   :mag:
:mag_right:   :unlock:   :lock:
:lock_with_ink_pen:   :closed_lock_with_key:   :key:
```

```
:bulb:   :flashlight:   :high_brightness:
:low_brightness:   :electric_plug:   :battery:
:calling:   :email:   :mailbox:
:postbox:   :bath:   :bathtub:
:shower:   :toilet:   :wrench:
:nut_and_bolt:   :hammer:   :seat:
:moneybag:   :yen:   :dollar:
:pound:   :euro:   :credit_card:
:money_with_wings:   :email:   :inbox_tray:
:outbox_tray:   :envelope:   :incoming_envelope:
:postal_horn:   :mailbox_closed:   :mailbox_with_mail:
:mailbox_with_no_mail:   :door:   :smoking:
:bomb:   :gun:   :hocho:
:pill:   :syringe:   :page_facing_up:
:page_with_curl:   :bookmark_tabs:   :bar_chart:
:chart_with_upwards_trend:   :chart_with_downwards_trend:   :scroll:
:clipboard:   :calendar:   :date:
:card_index:   :file_folder:   :open_file_folder:
:scissors:   :pushpin:   :paperclip:
:black_nib:   :pencil2:   :straight_ruler:
:triangular_ruler:   :closed_book:   :green_book:
:blue_book:   :orange_book:   :notebook:
:notebook_with_decorative_cover:   :ledger:   :books:
:bookmark:   :name_badge:   :microscope:
:telescope:   :newspaper:   :football:
:basketball:   :soccer:   :baseball:
:tennis:   :8ball:   :rugby_football:
:bowling:   :golf:   :mountain_bicyclist:
:bicyclist:   :horse_racing:   :snowboarder:
:swimmer:   :surfer:   :ski:
:spades:   :hearts:   :clubs:
:diamonds:   :gem:   :ring:
:trophy:   :musical_score:   :musical_keyboard:
:violin:   :space_invader:   :video_game:
:black_joker:   :flower_playing_cards:   :game_die:
:dart:   :mahjong:   :clapper:
:memo:   :pencil:   :book:
:art:   :microphone:   :headphones:
:trumpet:   :saxophone:   :guitar:
:shoe:   :sandal:   :high_heel:
:lipstick:   :boot:   :shirt:
:tshirt:   :necktie:   :womans_clothes:
:dress:   :running_shirt_with_sash:   :jeans:
:kimono:   :bikini:   :ribbon:
:tophat:   :crown:   :womans_hat:
:mans_shoe:   :closed_umbrella:   :briefcase:
:handbag:   :pouch:   :purse:
:eyeglasses:   :fishing_pole_and_fish:   :coffee:
:tea:   :sake:   :baby_bottle:
:beer:   :beers:   :cocktail:
:tropical_drink:   :wine_glass:   :fork_and_knife:
:pizza:   :hamburger:   :fries:
:poultry_leg:   :meat_on_bone:   :spaghetti:
:curry:   :fried_shrimp:   :bento:
:sushi:   :fish_cake:   :rice_ball:
:rice_cracker:   :rice:   :ramen:
:stew:   :oden:   :dango:
:egg:   :bread:   :doughnut:
:custard:   :icecream:   :ice_cream:
```

```
:shaved_ice:   :birthday:   :cake:
:cookie:   :chocolate_bar:   :candy:
:lollipop:   :honey_pot:   :apple:
:green_apple:   :tangerine:   :lemon:
:cherries:   :grapes:   :watermelon:
:strawberry:   :peach:   :melon:
:banana:   :pear:   :pineapple:
:sweet_potato:   :eggplant:   :tomato:
:corn:

Places

:house:   :house_with_garden:   :school:
:office:   :post_office:   :hospital:
:bank:   :convenience_store:   :love_hotel:
:hotel:   :wedding:   :church:
:department_store:   :european_post_office:   :city_sunrise:
:city_sunset:   :japanese_castle:   :european_castle:
:tent:   :factory:   :tokyo_tower:
:japan:   :mount_fuji:   :sunrise_over_mountains:
:sunrise:   :stars:   :statue_of_liberty:
:bridge_at_night:   :carousel_horse:   :rainbow:
:ferris_wheel:   :fountain:   :roller_coaster:
:ship:   :speedboat:   :boat:
:sailboat:   :rowboat:   :anchor:
:rocket:   :airplane:   :helicopter:
:steam_locomotive:   :tram:   :mountain_railway:
:bike:   :aerial_tramway:   :suspension_railway:
:mountain_cableway:   :tractor:   :blue_car:
:oncoming_automobile:   :car:   :red_car:
:taxi:   :oncoming_taxi:   :articulated_lorry:
:bus:   :oncoming_bus:   :rotating_light:
:police_car:   :oncoming_police_car:   :fire_engine:
:ambulance:   :minibus:   :truck:
:train:   :station:   :train2:
:bullettrain_front:   :bullettrain_side:   :light_rail:
:monorail:   :railway_car:   :trolleybus:
:ticket:   :fuelpump:   :vertical_traffic_light:
:traffic_light:   :warning:   :construction:
:beginner:   :atm:   :slot_machine:
:busstop:   :barber:   :hotsprings:
:checkered_flag:   :crossed_flags:   :izakaya_lantern:
:moyai:   :circus_tent:   :performing_arts:
:round_pushpin:   :triangular_flag_on_post:   :jp:
:kr:   :cn:   :us:
:fr:   :es:   :it:
:ru:   :gb:   :uk:
:de:

Symbols

:one:   :two:   :three:
:four:   :five:   :six:
:seven:   :eight:   :nine:
:keycap_ten:   :1234:   :zero:
:hash:   :symbols:   :arrow_backward:
:arrow_down:   :arrow_forward:   :arrow_left:
:capital_abcd:   :abcd:   :abc:
:arrow_lower_left:   :arrow_lower_right:   :arrow_right:
```

```
:arrow_up:   :arrow_upper_left:   :arrow_upper_right:
:arrow_double_down:   :arrow_double_up:   :arrow_down_small:
:arrow_heading_down:   :arrow_heading_up:   :leftwards_arrow_with_hook:
:arrow_right_hook:   :left_right_arrow:   :arrow_up_down:
:arrow_up_small:   :arrows_clockwise:   :arrows_counterclockwise:
:rewind:   :fast_forward:   :information_source:
:ok:   :twisted_rightwards_arrows:   :repeat:
:repeat_one:   :new:   :top:
:up:   :cool:   :free:
:ng:   :cinema:   :koko:
:signal_strength:   :u5272:   :u5408:
:u55b6:   :u6307:   :u6708:
:u6709:   :u6e80:   :u7121:
:u7533:   :u7a7a:   :u7981:
:sa:   :restroom:   :mens:
:womens:   :baby_symbol:   :no_smoking:
:parking:   :wheelchair:   :metro:
:baggage_claim:   :accept:   :wc:
:potable_water:   :put_litter_in_its_place:   :secret:
:congratulations:   :m:   :passport_control:
:left_luggage:   :customs:   :ideograph_advantage:
:cl:   :sos:   :id:
:no_entry_sign:   :underage:   :no_mobile_phones:
:do_not_litter:    :no_bicycles:
:no_pedestrians:   :children_crossing:   :no_entry:
:eight_spoked_asterisk:   :eight_pointed_black_star:   :heart_decoration:
:vs:   :vibration_mode:   :mobile_phone_off:
:chart:   :currency_exchange:   :aries:
:taurus:   :gemini:   :cancer:
:leo:   :virgo:   :libra:
:scorpius:   :sagittarius:   :capricorn:
:aquarius:   :pisces:   :ophiuchus:
:six_pointed_star:   :negative_squared_cross_mark:   :a:
:b:   :ab:   :o2:
:diamond_shape_with_a_dot_inside:   :recycle:   :end:
:on:   :soon:   :clock1:
:clock130:   :clock10:   :clock1030:
:clock11:   :clock1130:   :clock12:
:clock1230:   :clock2:   :clock230:
:clock3:   :clock330:   :clock4:
:clock430:   :clock5:   :clock530:
:clock6:   :clock630:   :clock7:
:clock730:   :clock8:   :clock830:
:clock9:   :clock930:   :heavy_dollar_sign:
:copyright:   :registered:   :tm:
:x:   :heavy_exclamation_mark:   :bangbang:
:interrobang:   :o:   :heavy_multiplication_x:
:heavy_plus_sign:   :heavy_minus_sign:   :heavy_division_sign:
:white_flower:   :100:   :heavy_check_mark:
:ballot_box_with_check:   :radio_button:   :link:
:curly_loop:   :wavy_dash:   :part_alternation_mark:
:trident:   :black_square:   :white_square:
:white_check_mark:   :black_square_button:   :white_square_button:
:black_circle:   :white_circle:   :red_circle:
:large_blue_circle:   :large_blue_diamond:   :large_orange_diamond:
:small_blue_diamond:   :small_orange_diamond:   :small_red_triangle:
:small_red_triangle_down:
```

This contains all of the emohi which can be used on GitHub.

When displayed in a `MarkdownTextView` or `MarkdownEditText` it should be shown with each emoji as a single character.
The Emoji are displayed as shown below:

Part 1	**Part 2**
	

There are some emoji which are not shown with a character not found symbol. This is a limitation of the font used rather than the emoji parsing, as the original emoji aliases have been removed.

Objective 9.ii.g: Checkboxes

The following sets of characters should be converted to unicode checkbox characters

- "[]" should be converted to ☐
- "[ ]" should be converted to ☐
- "[x]" should be converted to ☑

The checkbox test card contains the following which should be parsed to unicode ballot box characters:

- "[x]" at the start of the text immediately followed by an alphabet character

- “[]” on its own line
- “[ ]” at the start of a list line
- “[x]” at the start of a list line
- “[]” between two spaces within line
- “[x]” at the end of a line

And the following which should not:

- “[]” escaped ballot
- “[ ]” escaped ballot
- “[x]” escaped ballot

The test note is shown below

```
[x]Test for formatting checkboxes

Ballot on its own:

[]

Ballots at start of list

- [ ] Some text
- [x] Some more text


Ballot between text [] some more text

And the others [ ] and [x]


Escape ballots:
\[]
\[ ]
\[x]

Ending ballot:

[X]
```

Test	Expected result	Result
“[x]” at the start of the text	Unicode U+2611 character	Pass
“[]” on its own line	Unicode U+2610 character	Pass
“[ ]” at the start of a list line	Unicode U+2610 character	Pass

Test	Expected result	Result
“[x]” at the start of a list line	Unicode U+2611 character	Pass
“[]” between text in a line	Unicode U+2610 character	Pass
“[]”	“[]”	Pass
“[ ]”	“[ ]”	Pass
“[x]”	“[x]”	Pass
“[x]” at the end of the text	Unicode U+2611 character	Pass

The card was displayed as shown below:



☑Test for formatting checkboxes

Ballot on its own:

☐

Ballots at start of list

☐ Some text
☑ Some more text

Ballot between text ☐ some more text

And the others ☐ and ☑

Escape ballots: [] [ ] [x]

Ending ballot:

☑

Objective 9.ii.h: Text background colours

The test card for font background colours contains font tags with background-color attributes which should produce both white and black text colours.

The card is as follows

```
<font background-color="#000000">Test</font>

<font background-color="#FFFFFF">Test</font>
```

```
<font background-color="#7FFF00">Test</font>

<font background-color="#F0F8FF">Test</font>

<font background-color="#DC143C">Test</font>

<font background-color="#00FFFF">Test</font>

<font background-color="#0000FF">Test</font>

<font background-color="#2F4F4F">Test</font>

<font background-color="#FF1493">Test</font>

<font background-color="#7CFC00">Test</font>
```

Test	Expected Result	Result
Black background	White text	Pass
White background	Black text	Pass
Other colours	Text colour which ensures text is legible	Pass

The HTML above produced the following set of background and text colours in a `MarkdownEditText`:



Test

Test

Test

Test

Test

Test

Test

Test

Test

Test

The first two objectives are clearly met, and the third is also met as all of the text is legible.

Objective 9.b.i: Table placeholders

Tables are to be displayed as placeholders in the same way as large code blocks. They should show a table dialog when clicked, and maintain markdown formatting with in each table element.

The test card for tables contains three tables which should be converted to placeholder spans:

- A table at the start of the text
- A table within the text
- A table at the end of the text

And two tables which should not be converted:

- A table within a short code block
- A table within a large code block

The test card is below:

~~~
| Header 1 | Header 2 | Header 3 |

| — | — | — |

| Item 1 | Item 2 | Item 3 |

| **bold** | :smiley: | ~~strikethrough~~ |

Table in body text:

| Header 1 | Header 2 | Header 3 |

| — | — | — |

| Item 1 | Item 2 | Item 3 |

| **bold** | :smiley: | ~~strikethrough~~ |

Table in code span:

```
| Header 1 | Header 2 | Header 3 |
| — | — | — |
| Item 1 | Item 2 | Item 3 |
| bold | :smiley: | ~~strikethrough~~ |

```


Table rows duplicated to fill large code span:

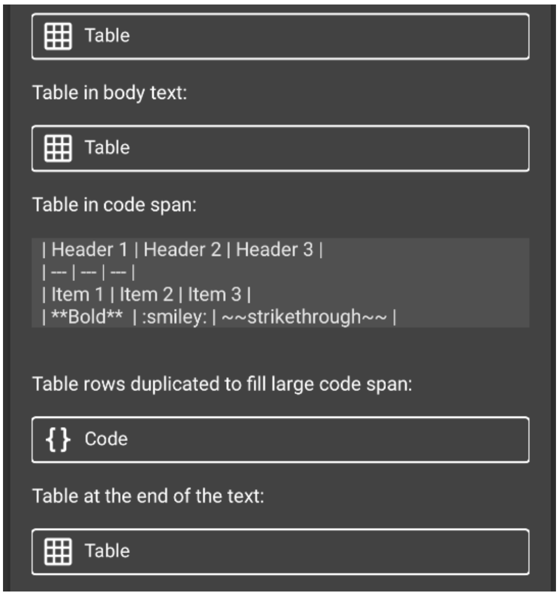
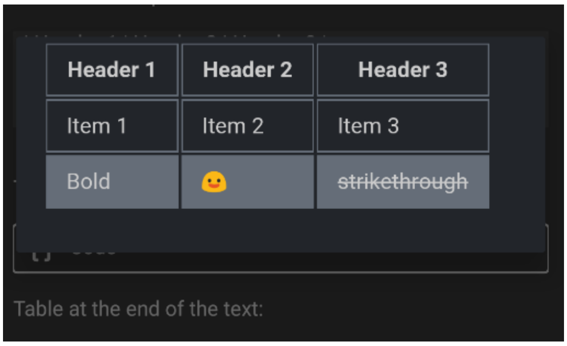
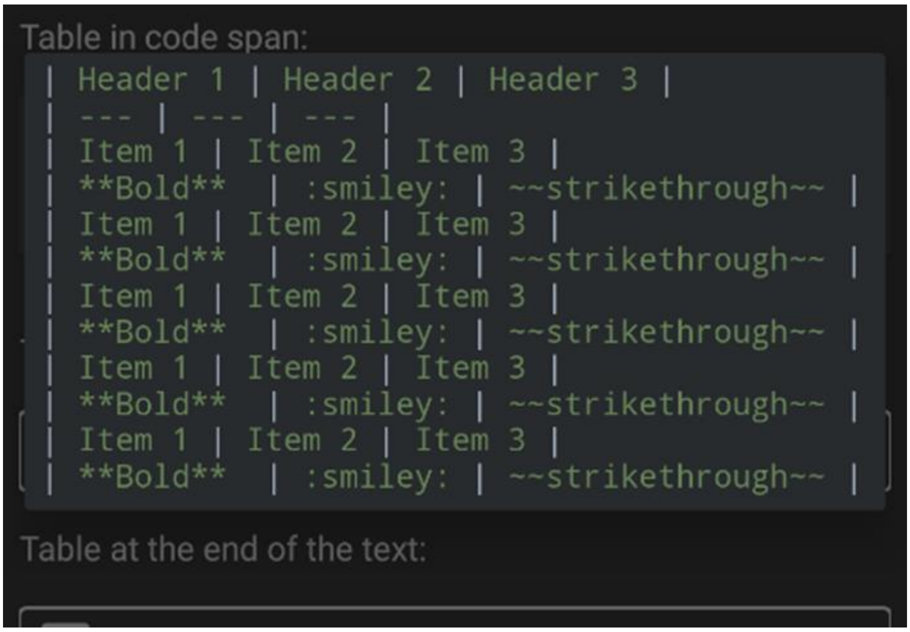
```
| Header 1 | Header 2 | Header 3 |
| --- | --- | --- |
| Item 1 | Item 2 | Item 3 |
| **Bold**  | :smiley: | ~~strikethrough~~ |
| Item 1 | Item 2 | Item 3 |
| **Bold**  | :smiley: | ~~strikethrough~~ |
| Item 1 | Item 2 | Item 3 |
| **Bold**  | :smiley: | ~~strikethrough~~ |
| Item 1 | Item 2 | Item 3 |
| **Bold**  | :smiley: | ~~strikethrough~~ |
| Item 1 | Item 2 | Item 3 |
| **Bold**  | :smiley: | ~~strikethrough~~ |
~~~

~~~~
```

Table at the end of the text:

| Header 1 | Header 2 | Header 3 |
| — | — | — |
| Item 1 | Item 2 | Item 3 |
| bold | :smiley: | ~~strikethrough~~ |
~~~~

Test	**Expected result**	**Result**
Table at start of text	Table placeholder displayed	Pass
Table within the text body	Table placeholder displayed	Pass
Table in short code span	Table markdown displayed in short code span	Pass
Table in large code span	Table markdown displayed when code span is clicked	Pass
Table at end of text	Table placeholder displayed	Pass
Emoji in table	Emoji displayed in table item	Pass
Strikethrough in table	Strikethrough displayed in table item	Pass

The card is displayed as shown below:

<table>
<tr><th>Description</th><th>Screenshot</th></tr>
<tr><td><b>Table spans</b></td><td> Table Table in body text: Table Table in code span: | Header 1	Header 2	Header 3
Item 1	Item 2	Item 3
Bold	:smiley:	~~strikethrough~~ | Table rows duplicated to fill large code span: { } Code Table at the end of the text: Table</td></tr>
<tr><td><b>Emoji in table</b></td><td><table><tr><th>Header 1</th><th>Header 2</th><th>Header 3</th></tr><tr><td>Item 1</td><td>Item 2</td><td>Item 3</td></tr><tr><td>Bold</td><td>😀</td><td>~~strikethrough~~</td></tr></table>Table at the end of the text:</td></tr>
<tr><td><b>Table formatting in code spans</b></td><td> Table in code span: | Header 1	Header 2	Header 3
Item 1	Item 2	Item 3
Bold	:smiley:	~~strikethrough~~
Item 1	Item 2	Item 3
Bold	:smiley:	~~strikethrough~~
Item 1	Item 2	Item 3
Bold	:smiley:	~~strikethrough~~
Item 1	Item 2	Item 3
Bold	:smiley:	~~strikethrough~~
Item 1	Item 2	Item 3
Bold	:smiley:	~~strikethrough~~ | Table at the end of the text:</td></tr>
</table>

Objective 9.c: Link handling

Part A- Matching URIs

Both URLs and emails should be matched, and converted into `URLSpans` to be handled by the Android system, or the app itself.
The following should be matched:

- URLs without and without a protocol
- Email addresses
- IP addresses with and without ports

URLs

URLs should be matched regardless of whether they contain the following parts:

- protocol
- path
- file type
- parameters
- fragment identifier

A URL containing all of these is `http://www.test.com/dir/filename.jpg?var1=foo#bar`

Email addresses

An email address consists of a local part, before the "@" symbol, and a domain after it.

The local part of the address may use the following characters:

- Alphanumeric
- Special characters !#$%&'*+-/=?^_\{|}~;`
- Dot "." if it is not the first or last character and does not appear consecutively unless quoted
- Space and "(),:;<>@[\] are allowed but not inside a quoted string
- Comments can be added at the start or end of the email address within a pair of brackets "()"

The domain part of an email address must match the requirements for a hostname:

- A list of dot separated DNS labels, each limited to 63 characters and consisting of the following characters:
    - Alphanumeric
    - Hyphen, “-”, at any position other than the first or last

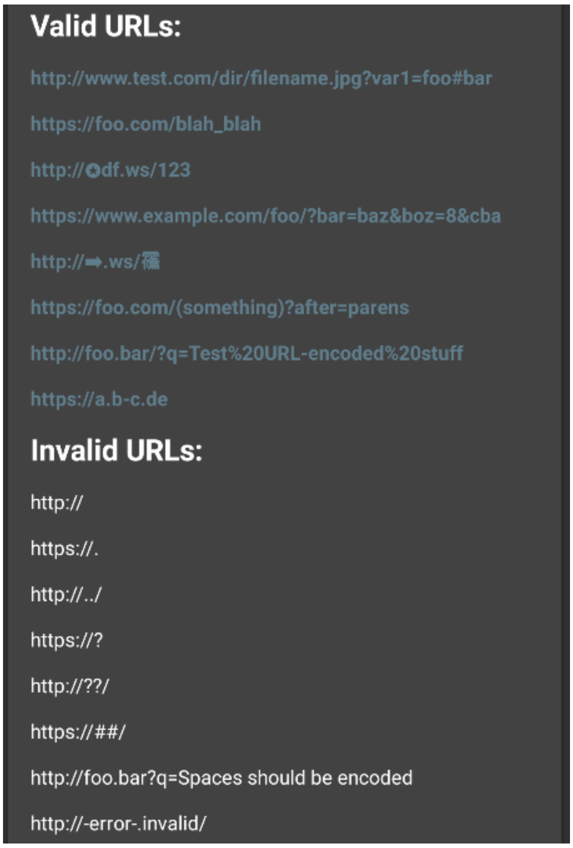
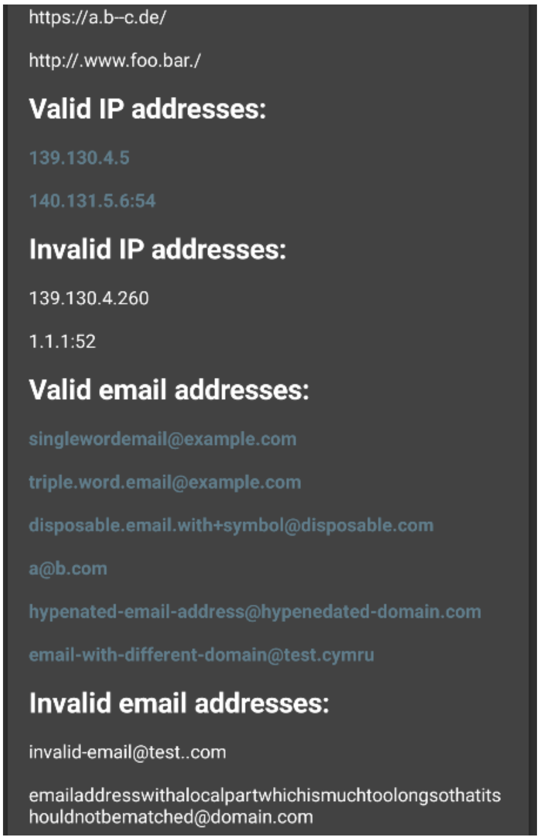
The test card for URL, email address, and IP address matching consists of the following:

```
# Valid URLs:

http://www.test.com/dir/filename.jpg?var1=foo#bar

https://foo.com/blah_blah

http://✪df.ws/123

https://www.example.com/foo/?bar=baz&boz=8&cba

http://➡.ws/䨹

https://foo.com/(something)?after=parens

http://foo.bar/?q=Test%20URL-encoded%20stuff

https://a.b-c.de

# Invalid URLs:

http://

https://.

http://../

https://?

http://??/

https://##/

http://foo.bar?q=Spaces should be encoded

http://-error-.invalid/

https://a.b--c.de/

http://.www.foo.bar./

# Valid IP addresses:

139.130.4.5

140.131.5.6:54

# Invalid IP addresses:
```

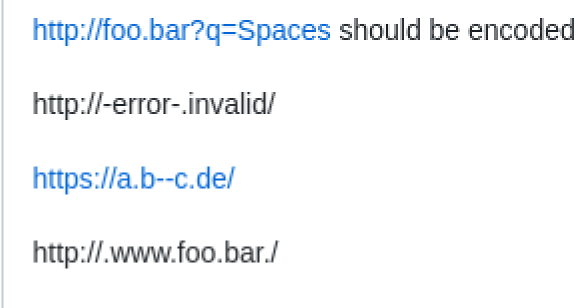
```
139.130.4.260

1.1.1:52

# Valid email addresses:

singlewordemail@example.com

triple.word.email@example.com

disposable.email.with+symbol@disposable.com

a@b.com

hypenated-email-address@hypenedated-domain.com

email-with-different-domain@test.cymru

# Invalid email addresses:

invalid-email@test..com

emailaddresswithalocalpartwhichismuchtoolongsothatitshouldnotbematched@domain.com
```

It should be self-explanatory that the lines which items are expected to be matched, and which are not.

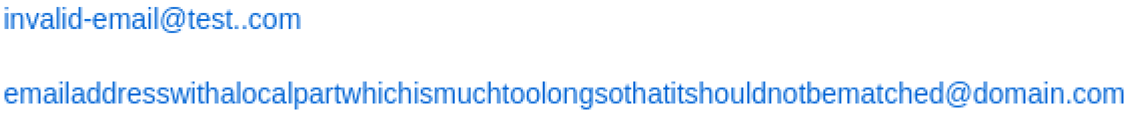
All of the tests were passed as expected:

Part 1	**Part 2**
 **Valid URLs:** http://www.test.com/dir/filename.jpg?var1=foo#bar https://foo.com/blah_blah http://✪df.ws/123 https://www.example.com/foo/?bar=baz&boz=8&cba http://➡.ws/䨹 https://foo.com/(something)?after=parens http://foo.bar/?q=Test%20URL-encoded%20stuff https://a.b-c.de **Invalid URLs:** http:// https://. http://../ https://? http://??/ https://##/ http://foo.bar?q=Spaces should be encoded http://-error-.invalid/	 https://a.b--c.de/ http://.www.foo.bar./ **Valid IP addresses:** 139.130.4.5 140.131.5.6:54 **Invalid IP addresses:** 139.130.4.260 1.1.1:52 **Valid email addresses:** singlewordemail@example.com triple.word.email@example.com disposable.email.with+symbol@disposable.com a@b.com hypenated-email-address@hypenedated-domain.com email-with-different-domain@test.cymru **Invalid email addresses:** invalid-email@test..com emailaddresswithalocalpartwhichismuchtoolongsothatitshouldnotbematched@domain.com

In this case the app performed better than GitHub's own website, as GitHub matched two the invalid URLs and both of the invalid email addresses:



> http://foo.bar?q=Spaces should be encoded
>
> http://-error-.invalid/
>
> https://a.b--c.de/
>
> http://.www.foo.bar./



invalid-email@test..com

emailaddresswithalocalpartwhichismuchtoolongsothatitshouldnotbematched@domain.com

Part B- Ignoring code

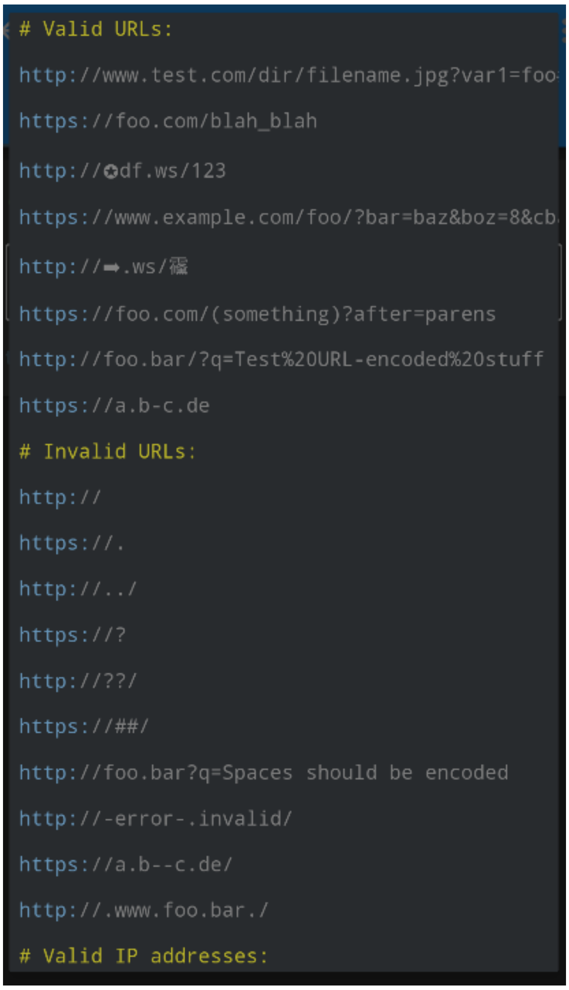
Many code segments, especially those in C style languages, contain code segments which might be matched as URLs.

Outside of code blocks, these should still be matched if they are valid URLs, such as “com.package.build”.

The test for ignoring these problematic strings involves wrapping the test used in part A within a code block. As such it is unnecessary to duplicate the body of text above in order to show another six characters.

As expected, the test was displayed correctly, as a code block placeholder.

This block contained the test in part A, without any modifications:



```
# Valid URLs:
http://www.test.com/dir/filename.jpg?var1=foo
https://foo.com/blah_blah
http://✪df.ws/123
https://www.example.com/foo/?bar=baz&boz=8&cb
http://➡.ws/䨹
https://foo.com/(something)?after=parens
http://foo.bar/?q=Test%20URL-encoded%20stuff
https://a.b-c.de
# Invalid URLs:
http://
https://.
http://../
https://?
http://??/
https://##/
http://foo.bar?q=Spaces should be encoded
http://-error-.invalid/
https://a.b--c.de/
http://.www.foo.bar./
# Valid IP addresses:
```

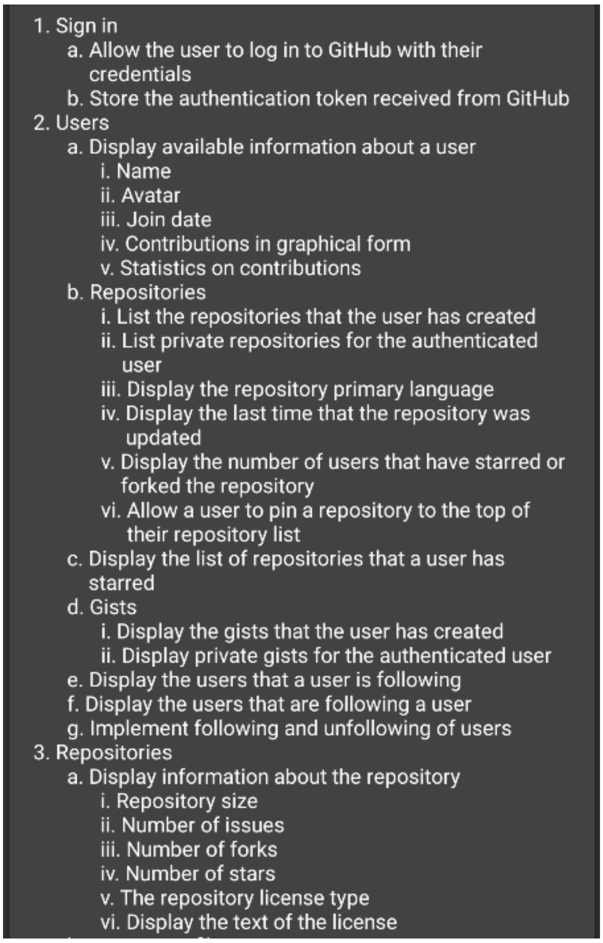
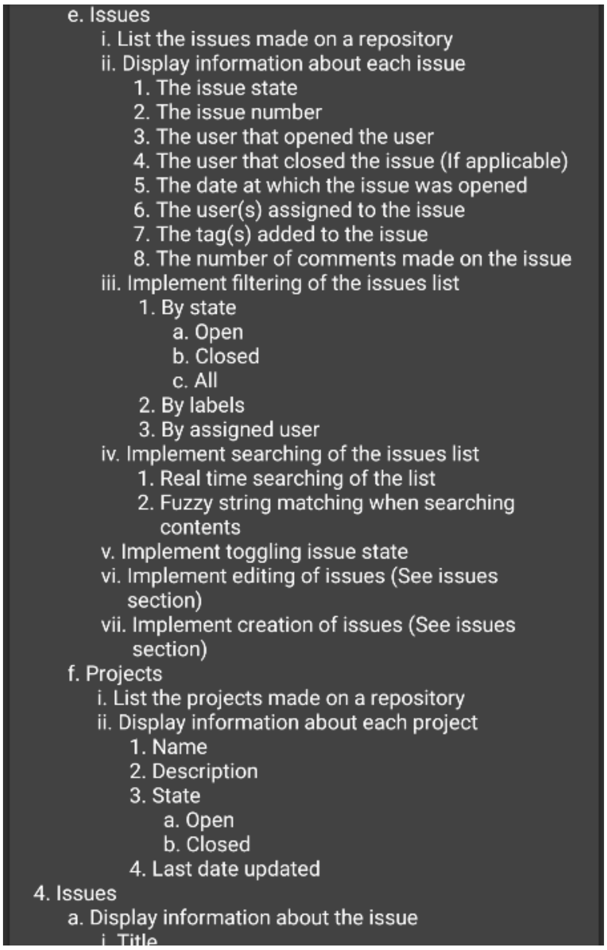
Objective 9.d: List formatting

In order to test this objective, I used three different lists declared in different ways.

Nested HTML list

The first is the list of objectives for this project which is declared as HTML and contains 32 ordered lists and 280 list items.
Each of the ordered lists has a type specified.

The app handled this as expected, displaying each nested element with the correct indentation and list item styling.

Part 1	**Part 2**
	

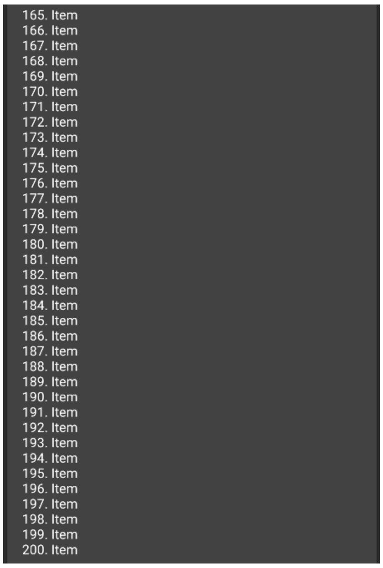
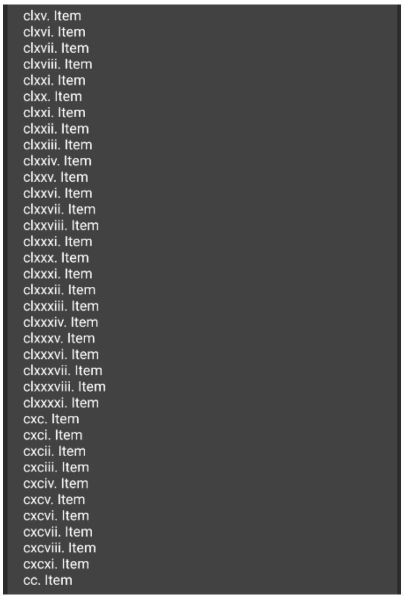
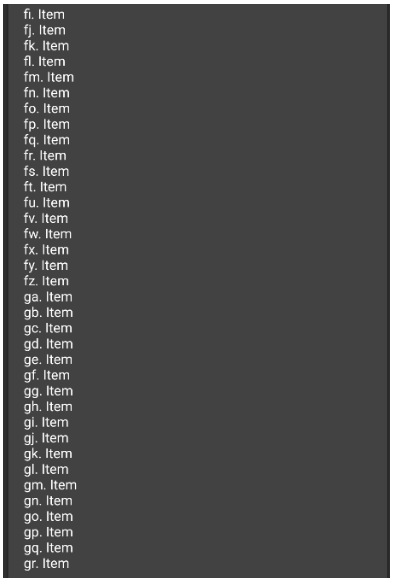
```
1. Sign in
    a. Allow the user to log in to GitHub with their
       credentials
    b. Store the authentication token received from GitHub
2. Users
    a. Display available information about a user
        i. Name
        ii. Avatar
        iii. Join date
        iv. Contributions in graphical form
        v. Statistics on contributions
    b. Repositories
        i. List the repositories that the user has created
        ii. List private repositories for the authenticated
            user
        iii. Display the repository primary language
        iv. Display the last time that the repository was
            updated
        v. Display the number of users that have starred or
           forked the repository
        vi. Allow a user to pin a repository to the top of
            their repository list
    c. Display the list of repositories that a user has
       starred
    d. Gists
        i. Display the gists that the user has created
        ii. Display private gists for the authenticated user
    e. Display the users that a user is following
    f. Display the users that are following a user
    g. Implement following and unfollowing of users
3. Repositories
    a. Display information about the repository
        i. Repository size
        ii. Number of issues
        iii. Number of forks
        iv. Number of stars
        v. The repository license type
        vi. Display the text of the license
```

```
    e. Issues
        i. List the issues made on a repository
        ii. Display information about each issue
            1. The issue state
            2. The issue number
            3. The user that opened the user
            4. The user that closed the issue (If applicable)
            5. The date at which the issue was opened
            6. The user(s) assigned to the issue
            7. The tag(s) added to the issue
            8. The number of comments made on the issue
        iii. Implement filtering of the issues list
            1. By state
                a. Open
                b. Closed
                c. All
            2. By labels
            3. By assigned user
        iv. Implement searching of the issues list
            1. Real time searching of the list
            2. Fuzzy string matching when searching
               contents
        v. Implement toggling issue state
        vi. Implement editing of issues (See issues
            section)
        vii. Implement creation of issues (See issues
             section)
    f. Projects
        i. List the projects made on a repository
        ii. Display information about each project
            1. Name
            2. Description
            3. State
                a. Open
                b. Closed
            4. Last date updated
4. Issues
    a. Display information about the issue
        i. Title
```

Long HTML list

The second list is intended to test handling of different list types for particularly long lists.
The single ordered list is declared in HTML and contains 200 list items.

It should be displayed up to 200 with the numeric type, cc in roman numerals, and gr in base 26.

Numeric	**Roman numerals**	**Base 26**
 165. Item 166. Item 167. Item 168. Item 169. Item 170. Item 171. Item 172. Item 173. Item 174. Item 175. Item 176. Item 177. Item 178. Item 179. Item 180. Item 181. Item 182. Item 183. Item 184. Item 185. Item 186. Item 187. Item 188. Item 189. Item 190. Item 191. Item 192. Item 193. Item 194. Item 195. Item 196. Item 197. Item 198. Item 199. Item 200. Item	 clxv. Item clxvi. Item clxvii. Item clxviii. Item clxxi. Item clxx. Item clxxi. Item clxxii. Item clxxiii. Item clxxiv. Item clxxv. Item clxxvi. Item clxxvii. Item clxxviii. Item clxxxi. Item clxxx. Item clxxxi. Item clxxxii. Item clxxxiii. Item clxxxiv. Item clxxxv. Item clxxxvi. Item clxxxvii. Item clxxxviii. Item clxxxxi. Item cxc. Item cxci. Item cxcii. Item cxciii. Item cxciv. Item cxcv. Item cxcvi. Item cxcvii. Item cxcviii. Item cxcxi. Item cc. Item	 fi. Item fj. Item fk. Item fl. Item fm. Item fn. Item fo. Item fp. Item fq. Item fr. Item fs. Item ft. Item fu. Item fv. Item fw. Item fx. Item fy. Item fz. Item ga. Item gb. Item gc. Item gd. Item ge. Item gf. Item gg. Item gh. Item gi. Item gj. Item gk. Item gl. Item gm. Item gn. Item go. Item gp. Item gq. Item gr. Item

All three of these tests passed successfully, and the formatting for particularly long roman numerals was better than GitHub which overflowed the bounds of its text box on some longer values such as “clxxxviii”.

Nested markdown list

This test is for a nested markdown list containing nested ordered and unordered lists, and list items containing other markdown elements.

The test markdown contains an unordered list, which contains:

- Two text items
- A text item with an emoji alias
- A nested unordered list which contains:
  - An inline code segment
- A nested ordered list which contains:
  - Two text items
  - An italicised text item
  - A bold text item
  - An italicised text item
  - A nested unordered check boxed list which contains:
    - A non-checked URL
    - A checked email address
    - A checked user reference

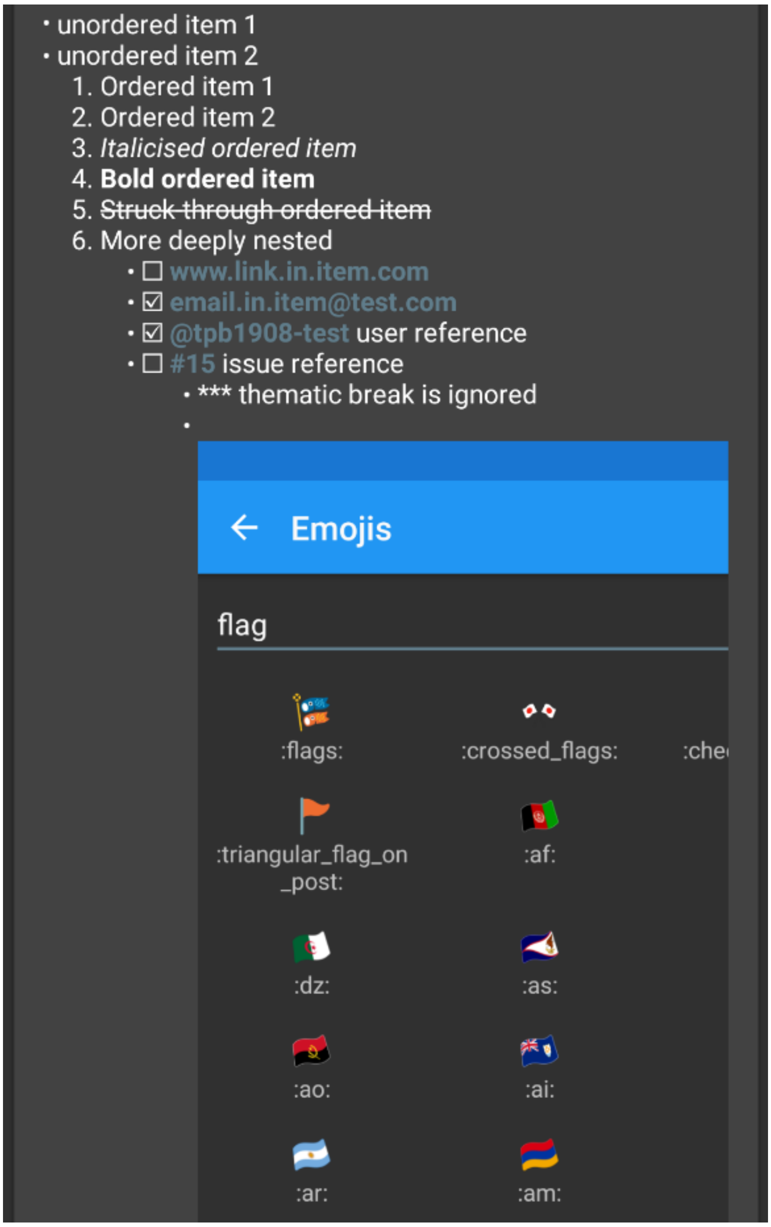
        - A non-checked issue reference
        - A nested unordered list which contains:
            - A triple asterisk which should be ignored
            - An image link which should be displayed in line

```
- unordered item 1
- unordered item 2
    1. Ordered item 1
    2. Ordered item 2
    3. *Italicised ordered item*
    4. **Bold ordered item**
    5. ~~Struck-through ordered item~~
    6. More deeply nested
        - [ ] www.link.in.item.com
        - [x] email.in.item@test.com
        - [x] @tpb1908-test user reference
        - [ ] #15 issue reference
            - *** thematic break is ignored
            - ![Image](http://imgur.com/Bap3G8x.png)
- :smiley: emoji in list item
    - ```Code in a nested item```
```

Test	**Expected result**	**Result**
Unordered text items	Displayed with correct indentation	Pass
Text item with emoji alias	Alias is displayed as its unicode character	Pass
Inline code segment	Inline code segment displayed with correct indentation	Pass
Nested lists	All child elements indented the same amount	Pass
Italicised text item	Text displayed in italics	Pass
Bold text item	Text displayed in bold	Pass
Strikethrough text item	Text displayed with strikethrough	Pass

Test	Expected result	Result
Checked list items	Unicode U+2611 checkbox displayed at start of item	Pass
Unchecked list items	Unicode U+2610 checkbox displayed at start of item	Pass
URL in list item	URL converted to clickable link	Pass
Email address in list item	Email address converted to clickable link	Pass
User reference in list item	User reference converted to clickable link	Pass
Issue reference in list item	Issue reference converted to clickable link	Pass
Triple asterisks in list item	Asterisks are not converted to horizontal rule	Pass
Image in list item	Image displayed in line with indentation	Pass

The markdown was displayed as shown below:



- unordered item 1
- unordered item 2
  1. Ordered item 1
  2. Ordered item 2
  3. *Italicised ordered item*
  4. **Bold ordered item**
  5. ~~Struck through ordered item~~
  6. More deeply nested
     - □ www.link.in.item.com
     - ☑ email.in.item@test.com
     - ☑ @tpb1908-test user reference
     - □ #15 issue reference
       - *** thematic break is ignored
       -

Objective 10: Markdown editing

It should already be clear form the “Markdown editing” section, that the interface for a markdown editor has been created, and that `Activities` for searching and selecting emoji and unicode characters have been implemented.
The purpose of this test section is to ensure that the flow for uploading an image can deal with the user exiting the process, or other problems occurring.

Test	Expected result	Result
The cancel button is clicked in the upload dialog	The dialog is cancelled and nothing is inserted	Pass
The take a picture button is clicked in the upload dialog	The camera is launched	Pass
The camera is cancelled without taking a picture	Nothing is inserted	Pass
A picture is taken	The camera closes and returns to the app	Pass
The choose from gallery button is clicked in the upload dialog	The default gallery application is launched	Pass
A picture is chosen from the gallery	The gallery closes and returns to the app	Pass
A valid image is returned from either the camera or gallery	The upload dialog is shown	Pass
There is no network connection	A suitable is shown when the user attempts to upload an image	Pass
The connection is lost while uploading the image	A suitable error message is shown	Pass

The app pass each of the tests, dealing with each of the ways that the user might attempt to upload an image or cancel doing so, and the problems which could occur while uploading the image.

Objective 8: Notifications

Notifications are more difficult to test than other features, as they must be triggered manually and then require waiting for a scheduled task to perform the required action.
Further, some notification types cannot be tested without being part of an organisation.

In order to test the most common notification types I used a test GitHub account to create comments and mentions on the test repository.

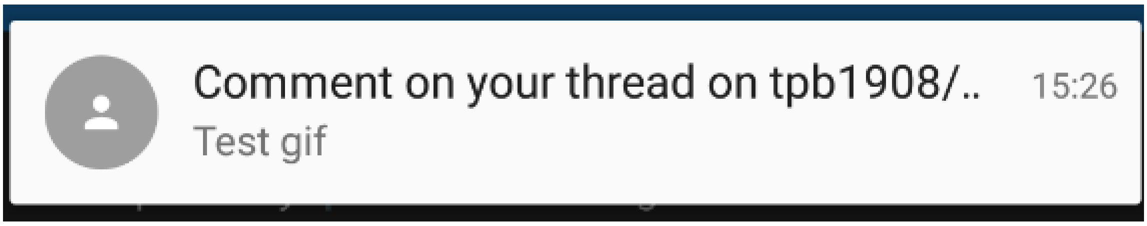
Issue comments

Issue comment notifications occur when another user comments on an issue which the authenticated user created.

In order to test this I used the test account to comment on an issue created by the authenticated account on the test repository.

Test	**Expected result**	**Result**
Comment made on issue	Notification received within 5 minutes	Pass
Notification displayed	Notification displayed with user icon	Pass
Notification clicked	Corresponding issue opened in app	Pass

The comment notification was displayed as shown below:



Comment on your thread on tpb1908/..

Test gif

15:26

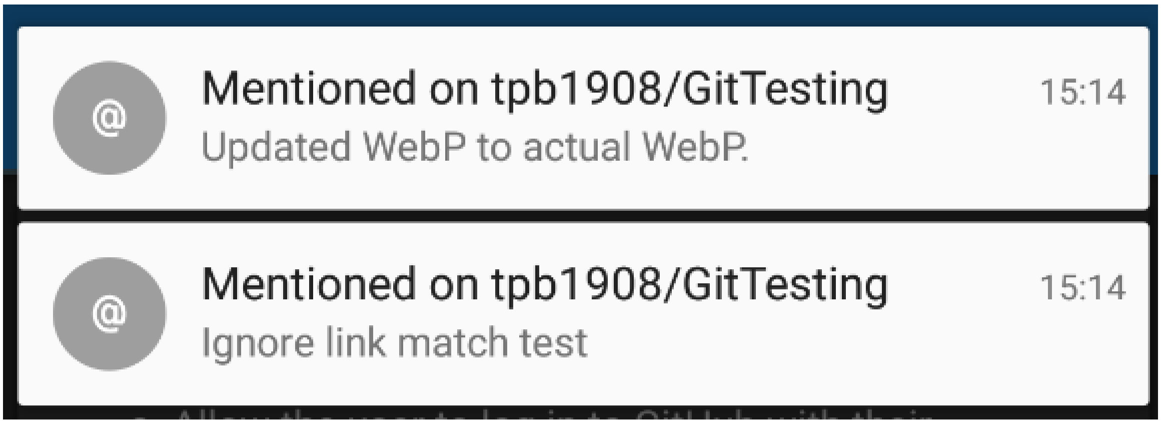
Mentions

Mentions occur when another user tags the authenticated user in a section of markdown.

In order to test notifications for mentions, I used the test account to mention my own account in an issue, and on a commit comment.

Test	Expected Result	Result
Authenticated user mentioned in issue	Notification received within 5 minutes	Pass
Authenticated user mentioned in commit comment	Notification received within 5 minutes	Pass
Mention notification received	"@" icon used	Pass
Issue mention notification clicked	Corresponding issue launched in app	Pass
Commit comment mention notification clicked	Corresponding commit launched in app	Pass

As I made the mentions close together, the notifications were received together and displayed as shown below:



Mentioned on tpb1908/GitTesting 15:14
Updated WebP to actual WebP.

Mentioned on tpb1908/GitTesting 15:14
Ignore link match test

Dismissing notifications

When notifications are opened or dismissed, the notification should be dismissed through the GitHub API.

Test	Expected Result	Result
Notification deleted	Dismiss `Intent` triggered	Pass
Notification dismiss `Intent` received	API call made, successfully dismissing notification	Pass

Proof of this test is best shown through the logs generated when sending the dismiss intent, and subsequently sending a network request to dismiss the notification.

The code used for generating the `PendingIntent` triggered when a notification is dismissed is as follows:

```
final Intent i = new
Intent(NotificationIntentService.this,NotificationIntentService.class);
i.setAction(ACTION_DELETE);
i.putExtra("notification", notif);
return PendingIntent.getService(this, 53253, i, PendingIntent.FLAG_ONE_SHOT);
```

When the `Intent` is received, the action will be ACTION_DELETE.
This is shown in the logs:

```
com.tpb.projects I/NotificationIntentService: onHandleIntent: Intent {
act=ACTION_DELETE
cmp=com.tpb.projects/.notifications.NotificationIntentService (has extras) }
```

The action is shown as ACTION_DELETE and the component to send the `Intent` to is the `NotificationIntentService` which originally created the `Intent`.
When the `Intent` is received in `onHandleIntent`, `markNotificationRead` is called, which calls the `Editor` method to make the request.
The log for the request sent is as follows:

```
com.tpb.projects I/LoggingInterceptor: Sending request
https://api.github.com/notifications/threads/omitted_id on
Connection{api.github.com:443, proxy=DIRECT@
hostAddress=api.github.com/192.30.253.117:443
cipherSuite=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 protocol=http/1.1}
                                       Accept: application/vnd.github.v3+json
                                       Authorization: token omitted_token
                                       Content-Type: application/x-www-form-
urlencoded

                                       Content-Length: 0
                                       Host: api.github.com
                                       Connection: Keep-Alive
                                       Accept-Encoding: gzip
                                       User-Agent: okhttp/3.6.0
```

This shows that the request is being sent to the correct API endpoint.

The log for the request response is as follows:

```
com.tpb.projects I/LoggingInterceptor: Received response for
https://api.github.com/notifications/threads/omitted_id in 609.8ms
                                       Server: GitHub.com
                                       Date: Sat, 29 Apr 2017 14:30:55 GMT
                                       Transfer-Encoding: chunked
                                       Status: 205 Reset Content
                                       X-RateLimit-Limit: 5000
                                       X-RateLimit-Remaining: 4987
                                       X-RateLimit-Reset: 1493479595
                                       X-OAuth-Scopes: gist, repo, user
                                       X-Accepted-OAuth-Scopes: notifications,
repo
                                       X-OAuth-Client-Id: omitted_client_id
                                       X-GitHub-Media-Type: github.v3;
format=json
                                       Access-Control-Expose-Headers: ETag,
Link, X-GitHub-OTP, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-
Reset, X-OAuth-Scopes, X-Accepted-OAuth-Scopes, X-Poll-Interval
                                       Access-Control-Allow-Origin: *
                                       Content-Security-Policy: default-src
'none'
                                       Strict-Transport-Security: max-
age=31536000; includeSubdomains; preload
                                       X-Content-Type-Options: nosniff
                                       X-Frame-Options: deny
                                       X-XSS-Protection: 1; mode=block
                                       X-GitHub-Request-Id:
DB9A:6F20:27DB76F:3335CCD:5904A39E
```

The response shows the 205 reset content status which is listed as the successful response for marking notifications as read.

Objective 7: Link handling

This objective is one of the simplest to test.

Throughout the development process of the app I have added content to the test repository and launched the app through a URL rather than navigating through the UI in order to save time.

Note: All links in data are prepended with https://github.com/

Test	Data	Expected Result	Result
Username link	tpb1908 tpb1908-test	Open the `UserActivity` with the specified user	Pass
Repository link	tpb1908/GitTesting	Open the GitTesting repository	Pass
Repository issues link	tpb1908/GitTesting/issues	Open the issues tab in the `RepositoryActivity`	Pass
Repository commits link	tpb1908/GitTesting/commits	Open the commits tab in the `RepositoryActivity`	Pass
Issue link	tpb1908/GitTesting/issues/5	Open the issue with number 5	Pass
Invalid issue link	tpb1908/GitTesting/issues/abc	Reject the link	Pass
Other users' issue link	tpb1908/GitTesting/issues/56	Open issue 56 with full access	Pass
Other user's locked issue link	tpb1908-test/Testing/issues/1	Open the issue without comment access	Pass
Commit link	tpb1908/GitTesting/commit/af60c3141a699362d582e668eca42937ab22459e	Open commit with hash	Pass
Project link	tpb1908/GitTesting/projects/1	Open the project with id 1	Pass
Project card	tpb1908/GitTesting/projects/1#card-2705834	Open the project with id 1 and highlight the card with id 2705834	Pass
Path link	tpb1908/AndroidProjectsClient/tree/master/app	Open `ContentActivity`	Pass

Test	Data	Expected Result	Result
File link	tpb1908/AndroidProjectsClient/blob/899d0bb4b3e5b3fcfad8b5b2fe404a53793940c9/app/src/main/java/com/tpb/projects/util/Interceptor.java	Open the `Interceptor` class	Pass

The app opened all of these links successfully, launching the correct `Activities` with the correct state.
Objective 7.ii also states that the app should gracefully reject unsupported links.

Test	Data	Expected Result	Result
Invalid link of length 3	tpb1908/GitTesting/commits	Chooser dialog shown	Pass
Invalid link of length 4	tpb1908/GitTesting/commits/somehash	Chooser dialog shown	Pass
Invalid link of greater length	tpb1908/GitTesting/some/path/which/is/not/valid	Chooser dialog shown	Pass