Contents

[**Testing** 1](#_Toc482007893)

[**Markdown** 1](#_Toc482007894)

[**Objective 9.ii.a: Code blocks** 1](#_Toc482007895)

[**Objective 9.ii.b: Horizontal rules** 6](#_Toc482007896)

[**Objective 9.ii.c: Image handling and relative link handling** 8](#_Toc482007897)

[**Objective 9.ii.d: Username linking** 9](#_Toc482007898)

[**Objective 9.ii.e: Issue linking** 11](#_Toc482007899)

[**Objective 9.ii.f: Emoji parsing** 12](#_Toc482007900)

[**Objective 9.ii.g: Checkboxes** 18](#_Toc482007901)

[**Objective 9.ii.h: Text background colours** 20](#_Toc482007902)

[**Objective 9.b.i: Table placeholders** 22](#_Toc482007903)

[**Objective 9.c: Link handling** 26](#_Toc482007904)

[**Objective 9.d: List formatting** 30](#_Toc482007905)

[**Objective 10: Markdown editing** 36](#_Toc482007906)

[**Objective 8: Notifications** 37](#_Toc482007907)

[**Issue comments** 37](#_Toc482007908)

[**Mentions** 37](#_Toc482007909)

[**Dismissing notifications** 38](#_Toc482007910)

[**Objective 7: Link handling** 40](#_Toc482007911)

**Testing**

**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

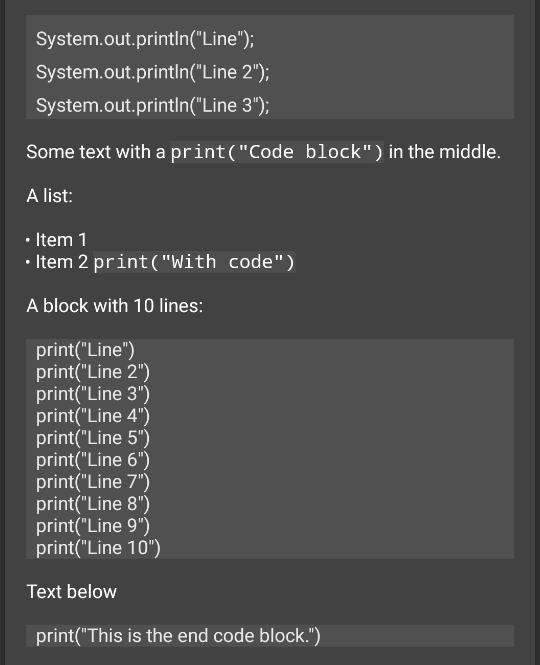
- Item 2 ```print(“With code”)```

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:



**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** |
| --- | --- |
| http://imgur.com/KFszbeL.png | http://imgur.com/yD7hjM3.png |

**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:

\\*\*\*

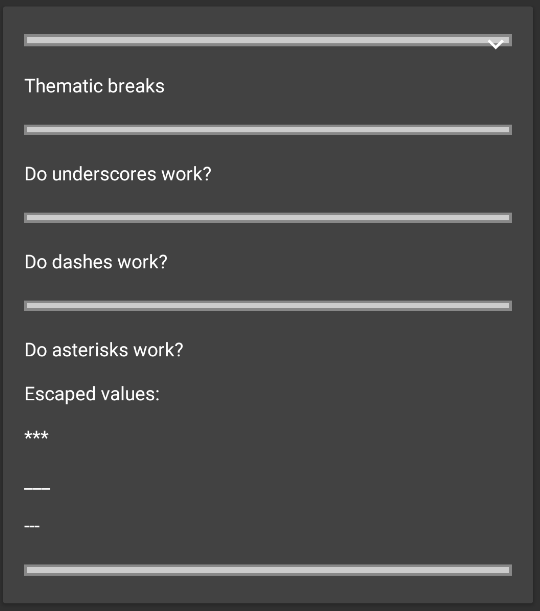
\\_\_\_

\—

—

| **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:



**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** |
| --- | --- |
| http://imgur.com/B4aqUDZ.png | http://imgur.com/ClujAd6.png |

**Objective 9.ii.d: Username linking**

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

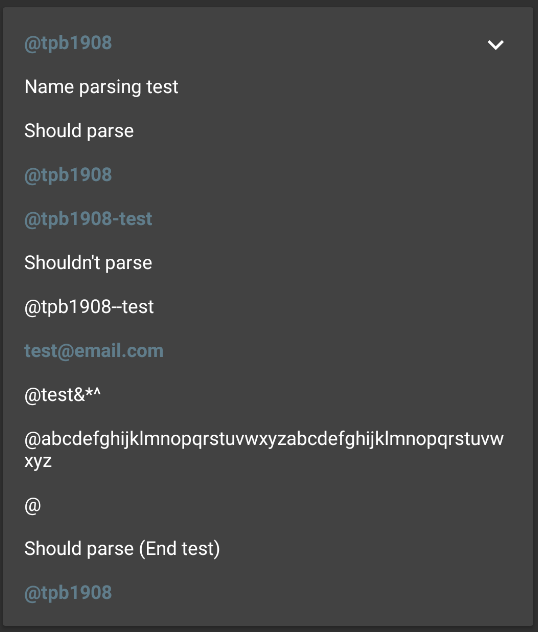
* Usernames linked at the start of the text
* Usernames with a single hyphen
* Usernames at the end of the text

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:



**Objective 9.ii.e: Issue linking**

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

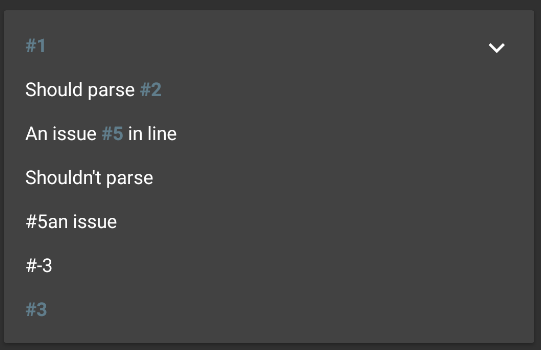
* An issue at the start of the text
* An issue on its own line
* An issue between two other blocks of text
* An issue at the end of the text

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 |
| 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:



**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** |
| --- | --- |
| http://imgur.com/NWrQwLE.png | http://imgur.com/zhDHYYm.png |

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:

\[]

\[ ]

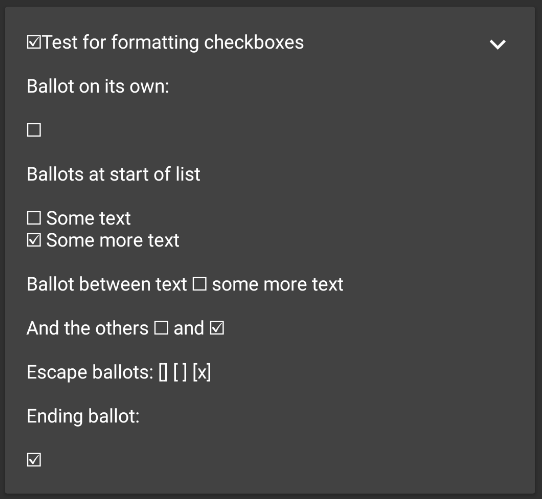
\[x]

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 |
| “[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:



**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>

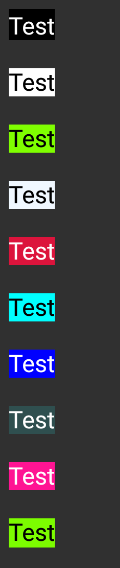
<font background-color="#2F4F4F">Test</font>

<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:



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:

| **Description** | **Screenshot** |
| --- | --- |
| **Table spans** | http://imgur.com/JvDAER4.png |
| **Emoji in table** | http://imgur.com/AsoZWqT.png |
| **Table formatting in code spans** |  |

**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

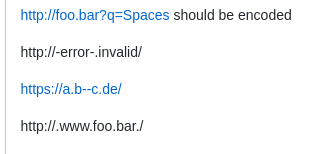
emailaddresswithalocalpartwhichismuchtoolongsothatitshouldnotbematched@domain.com

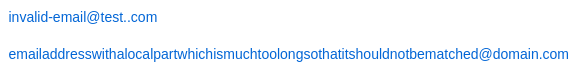
It should be self-explanatory that the lines which items are expected to be matched, and which are not.

All of the tests were passed as expected:

| **Part 1** | **Part 2** |
| --- | --- |
| http://imgur.com/T8Tf7wc.png | http://imgur.com/Ujnnpta.png |

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:





**Part B- Ignoring code**

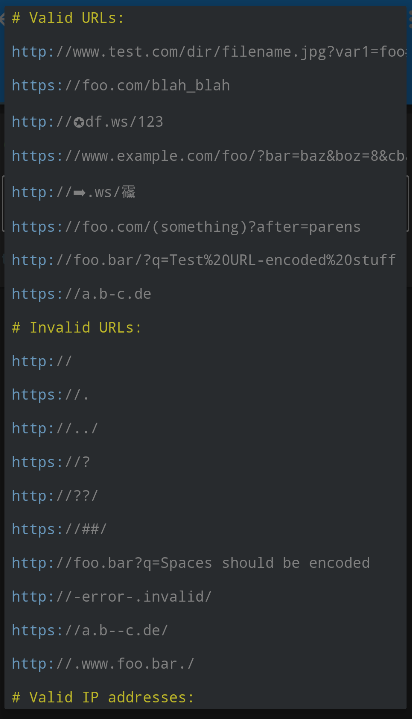
Many code segments, especially those in C style languages, contain code segments which might be matched as URLs.

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:



**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** |
| --- | --- |
| http://imgur.com/t1hp6of.png | http://imgur.com/dMexwLm.png |

**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** |
| --- | --- | --- |
| http://imgur.com/jUtaycB.png | http://imgur.com/IT2tJVX.png | http://imgur.com/VBn6cl9.png |

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

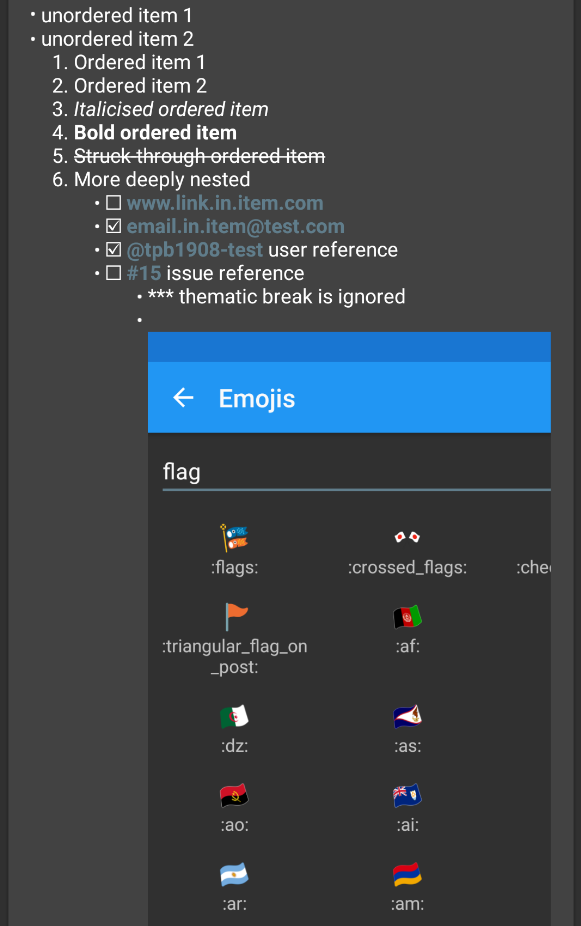
- ![Image](http://imgur.com/Bap3G8x.png)

- :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 |
| 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:



**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.

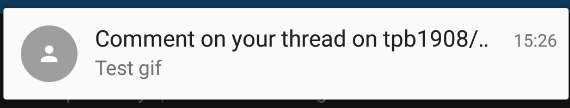
**Issue comments**

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:



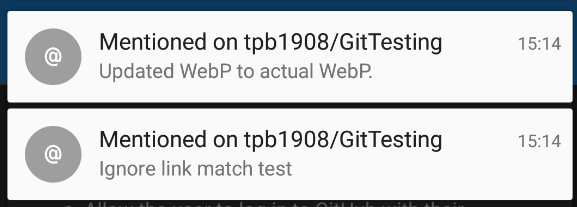
**Mentions**

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:



**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 succesfull 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 |
| 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/comits | Chooser dialog shown | Pass |
| Invalid link of length 4 | tpb1908/GitTesting/comits/somehash | Chooser dialog shown | Pass |
| Invalid link of greater length | tpb1908/GitTesting/some/path/which/is/not/valid | Chooser dialog shown | Pass |