

Mobile

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Thu Jun 11 20:09:16 2020 End: Thu Jun 11 20:58:36 2020 Duration: 0:49:19.891000  broken: 0 (0.00%) failed: 13 (24.53%) skipped: 0 (0.00%) passed: 40 (75.47%) |  |

Test Results

# Checking create playlist with empty name - failed

Checking create playlist with empty name

## Details

AssertionError: assert False

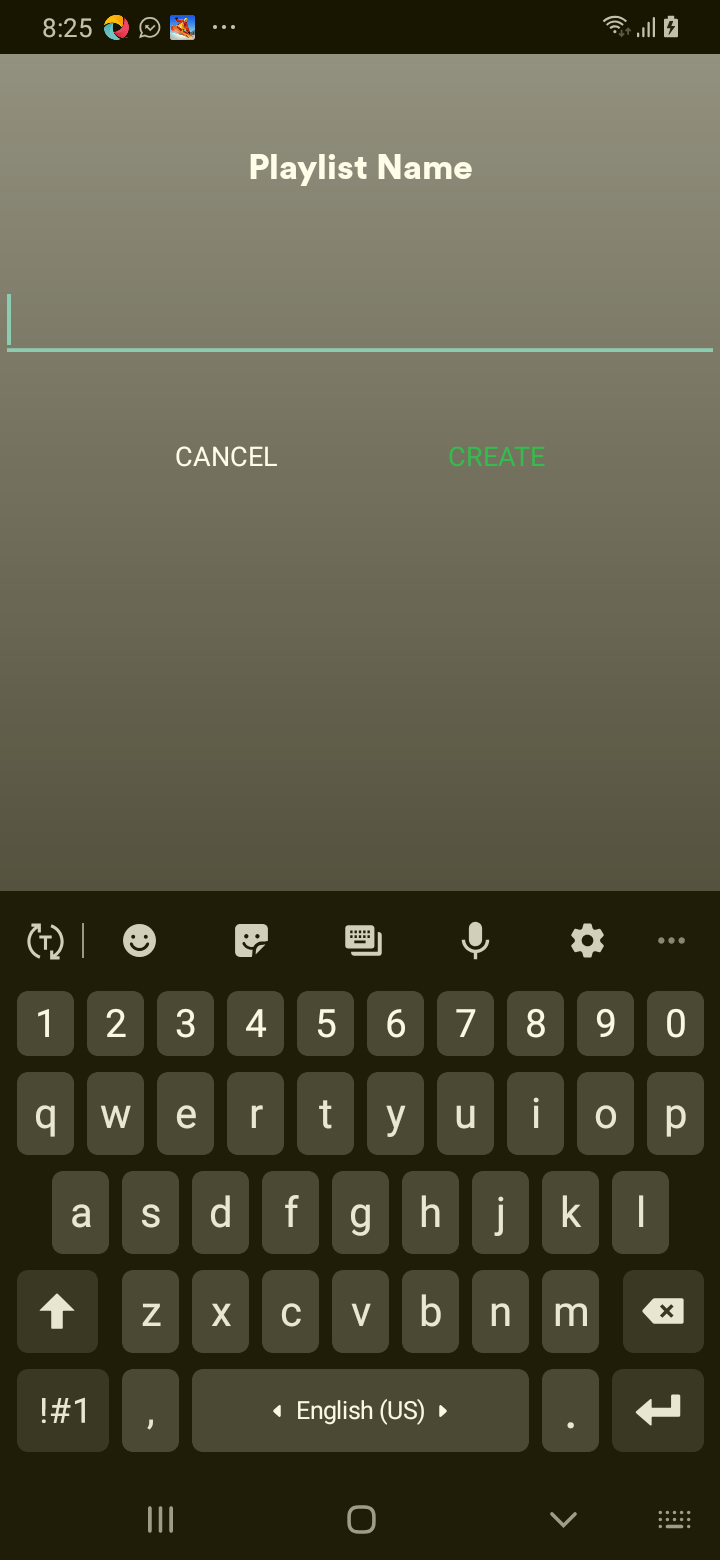
|  |
| --- |
| self = <test\_playlist.TestPlaylist object at 0x1072e38b0>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Playlist Tests")  @allure.sub\_suite("Checking create playlist with empty name")  @allure.title("Checking create playlist with empty name")  @allure.description("Checking create playlist with empty name")  @pytest.mark.Do  @pytest.mark.Playlist  @pytest.mark.Test15  def test\_case\_15(self, setup):  """  Checking create playlist with empty name  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_library\_button()  pl = PlayList(self.driver)  if not pl.create\_new\_playlist(""):  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "Creation done")  assert False  if pl.check\_create\_button():  Helper.report\_allure(self.driver, "Creation failed")  assert True  else:  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "Creation done") > assert False E assert False  Mobile\_Testing/Tests/test\_playlist.py:504: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Creation done



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user email to !@!#@ email - failed

changes the user email to !@!#@ email

## Details

AssertionError: assert False

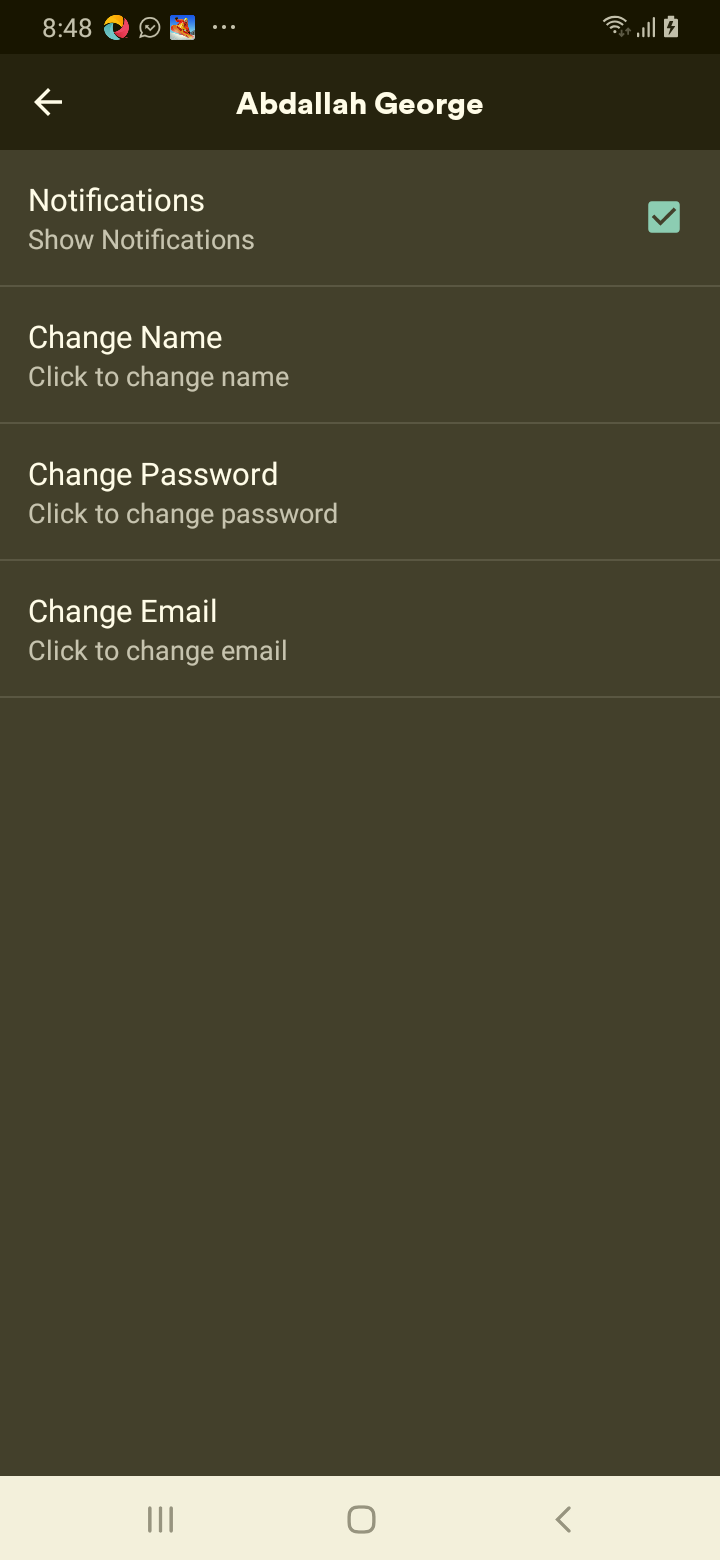
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d185a00>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user email to !@!#@ email")  @allure.title("changes the user email to !@!#@ email")  @allure.description("changes the user email to !@!#@ email")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test19  def test\_case\_19(self, setup):  """  changes the user email to !@!#@ email  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to !@!#@ email")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_the\_email("!@!#@ ","or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_the\_email("abdallah@gmail.com", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to !@!#@ email") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:780: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user email to !@!#@ email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user email to empty email - failed

changes the user email to empty email

## Details

AssertionError: assert False

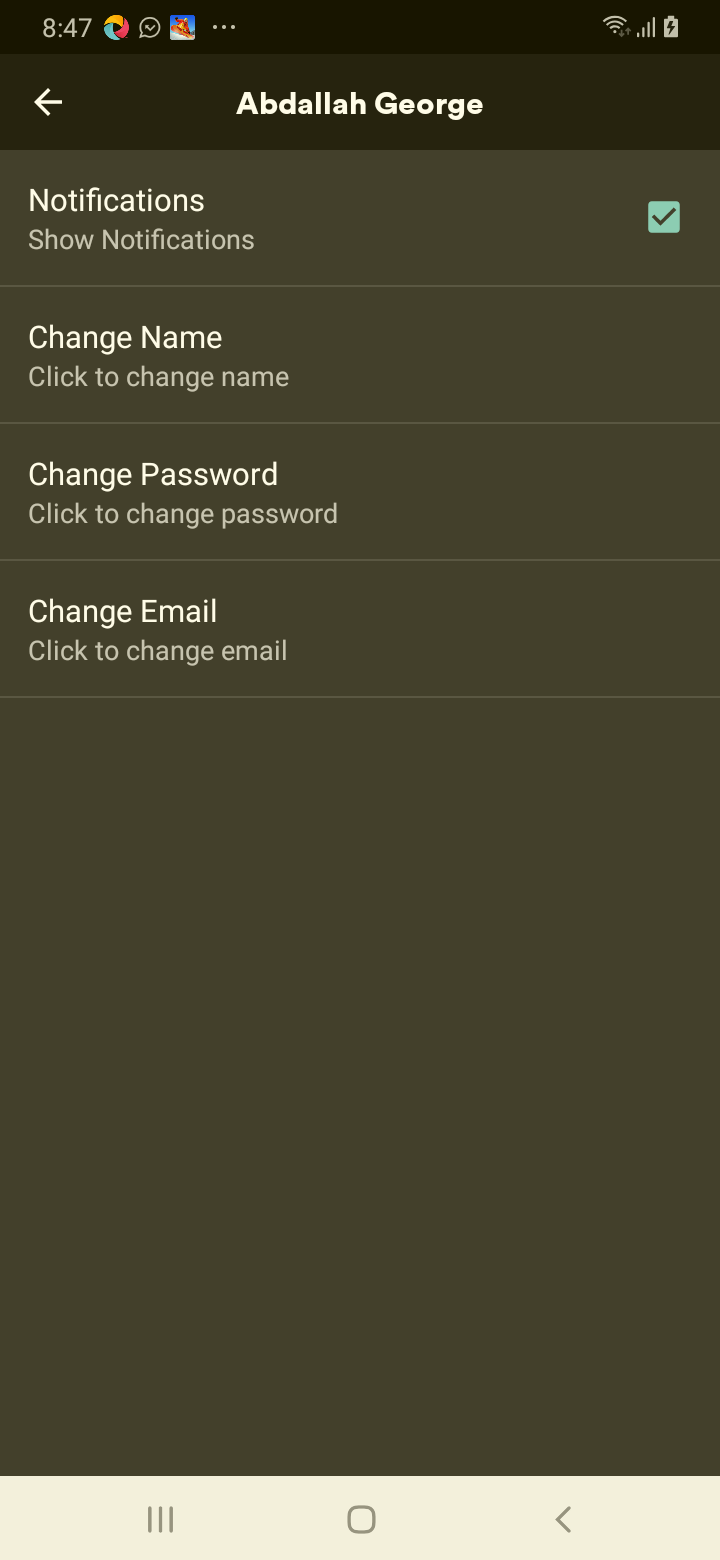
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d2c87c0>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user email to empty email")  @allure.title("changes the user email to empty email")  @allure.description("changes the user email to empty email")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test18  def test\_case\_18(self, setup):  """  changes the user email to empty email  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to empty email")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_the\_email("", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_the\_email("abdallah@gmail.com", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to empty email") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:743: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user email to empty email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user email to numbers email - failed

changes the user email to numbers email

## Details

AssertionError: assert False

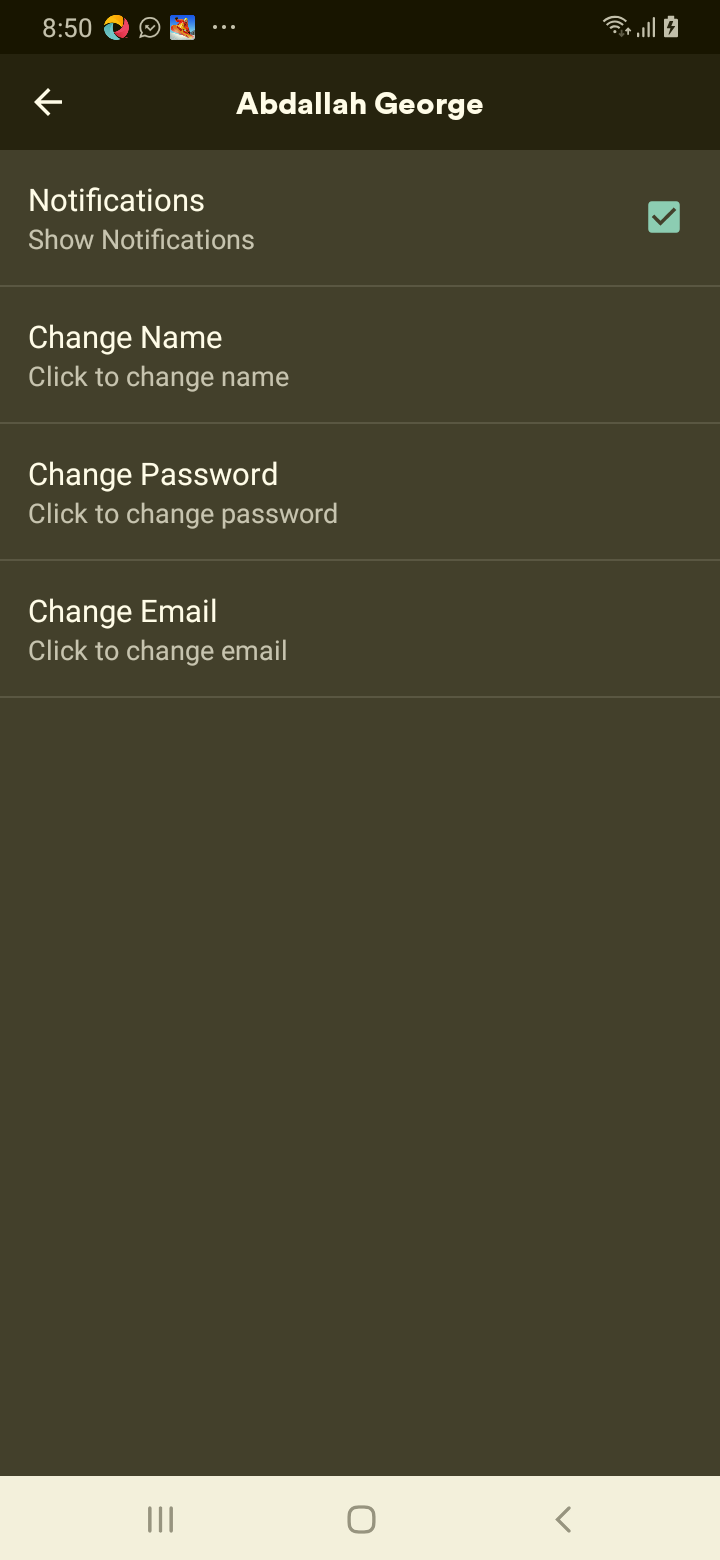
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d3c6d30>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user email to numbers email")  @allure.title("changes the user email to numbers email")  @allure.description("changes the user email to numbers email")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test21  def test\_case\_21(self, setup):  """  changes the user email to numbers email  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to numbers email")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_the\_email("1234567", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_the\_email("abdallah@gmail.com", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to numbers email") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:854: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user email to numbers email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user email to spaces email - failed

changes the user email to spaces email

## Details

AssertionError: assert False

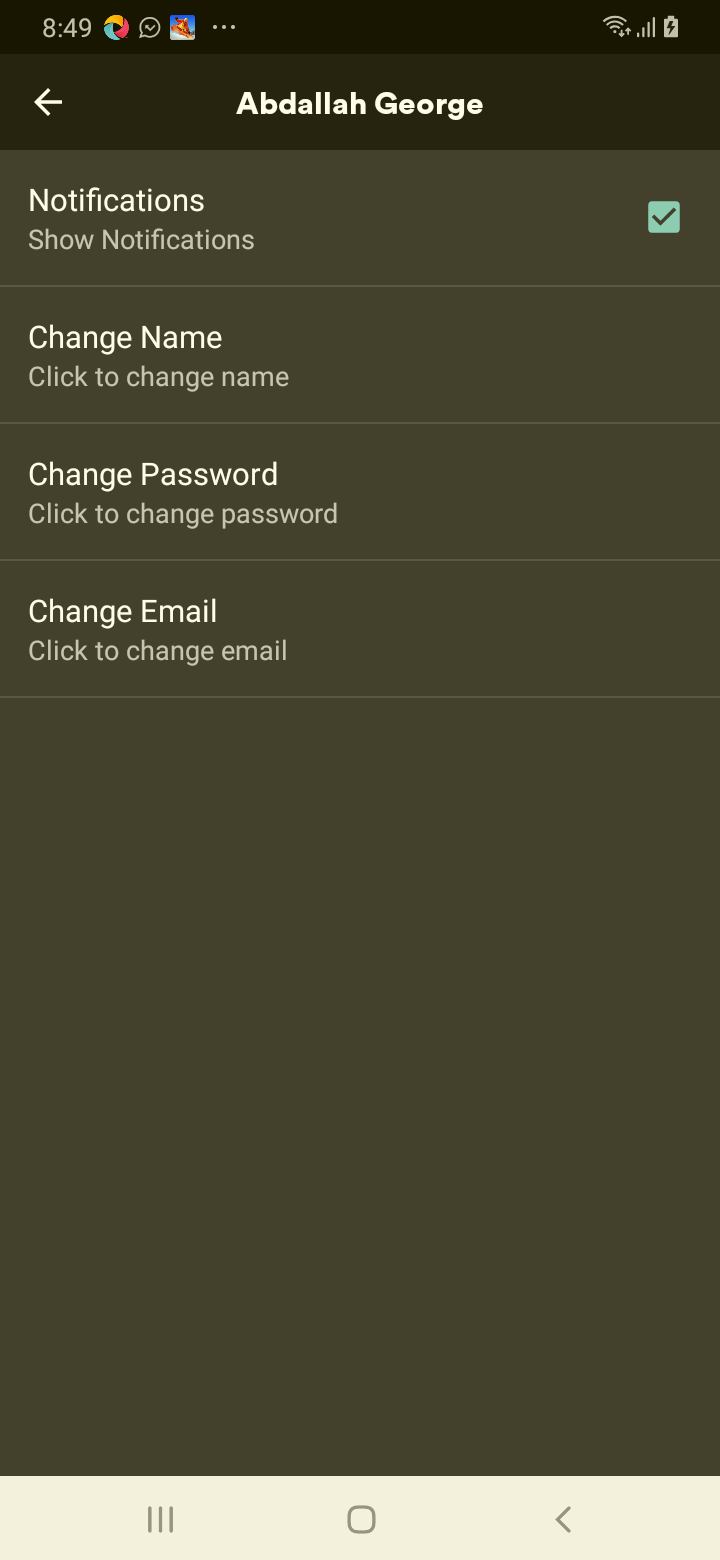
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10bff3b50>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user email to spaces email")  @allure.title("changes the user email to spaces email")  @allure.description("changes the user email to spaces email")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test20  def test\_case\_20(self, setup):  """  changes the user email to spaces email  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to spaces email")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_the\_email(" ", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_the\_email("abdallah@gmail.com", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user email to spaces email") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:817: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user email to spaces email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user name to !@!#@ name - failed

changes the user name to !@!#@ name

## Details

AssertionError: assert False

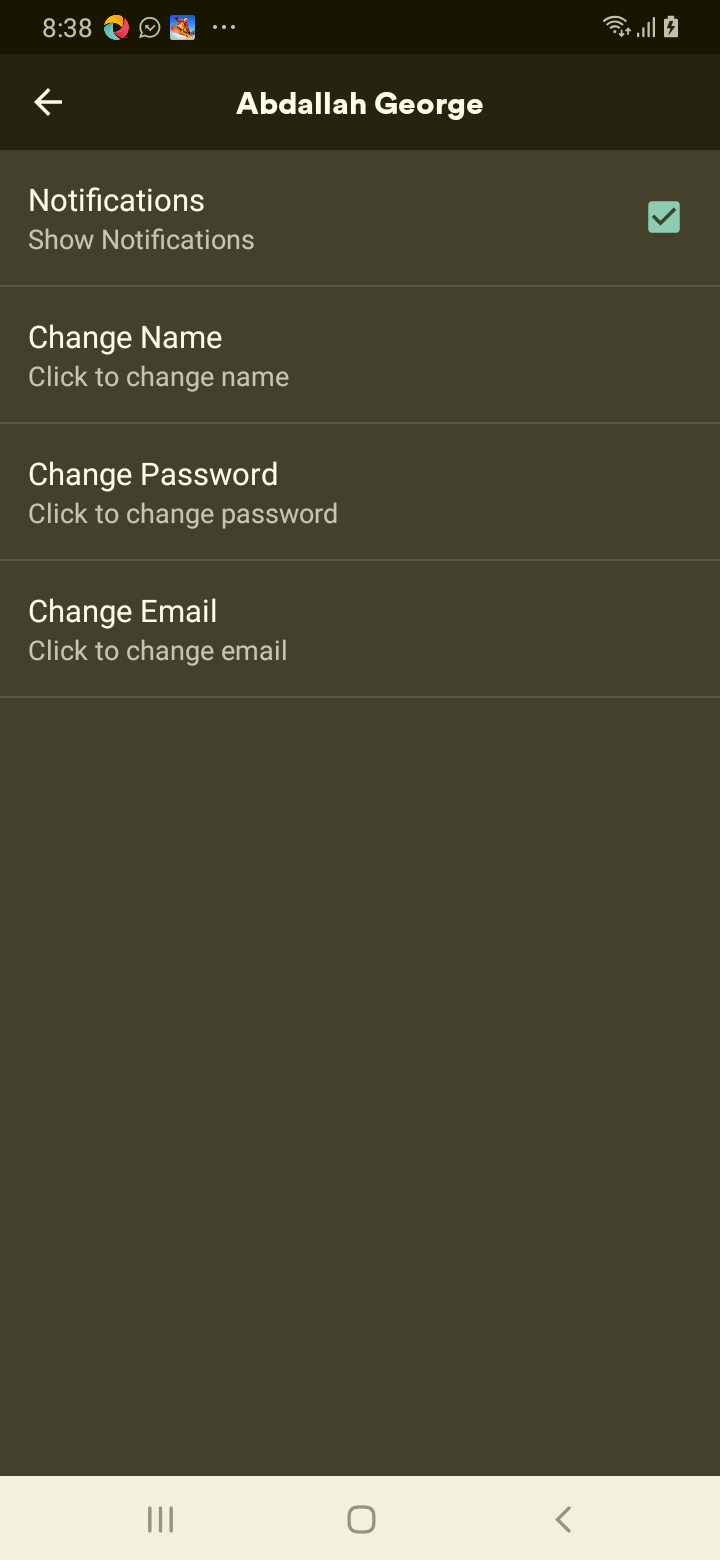
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d17be50>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user name to !@!#@ name")  @allure.title("changes the user name to !@!#@ name")  @allure.description("changes the user name to !@!#@ name")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test9  def test\_case\_9(self, setup):  """  changes the user name to !@!#@ name  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user name to !@!#@ name")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_name("!@!#@", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_name("Abdallah George", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test changes the user name to !@!#@ name") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:412: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test changes the user name to !@!#@ name



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user name to empty name - failed

changes the user name to empty name

## Details

AssertionError: assert False

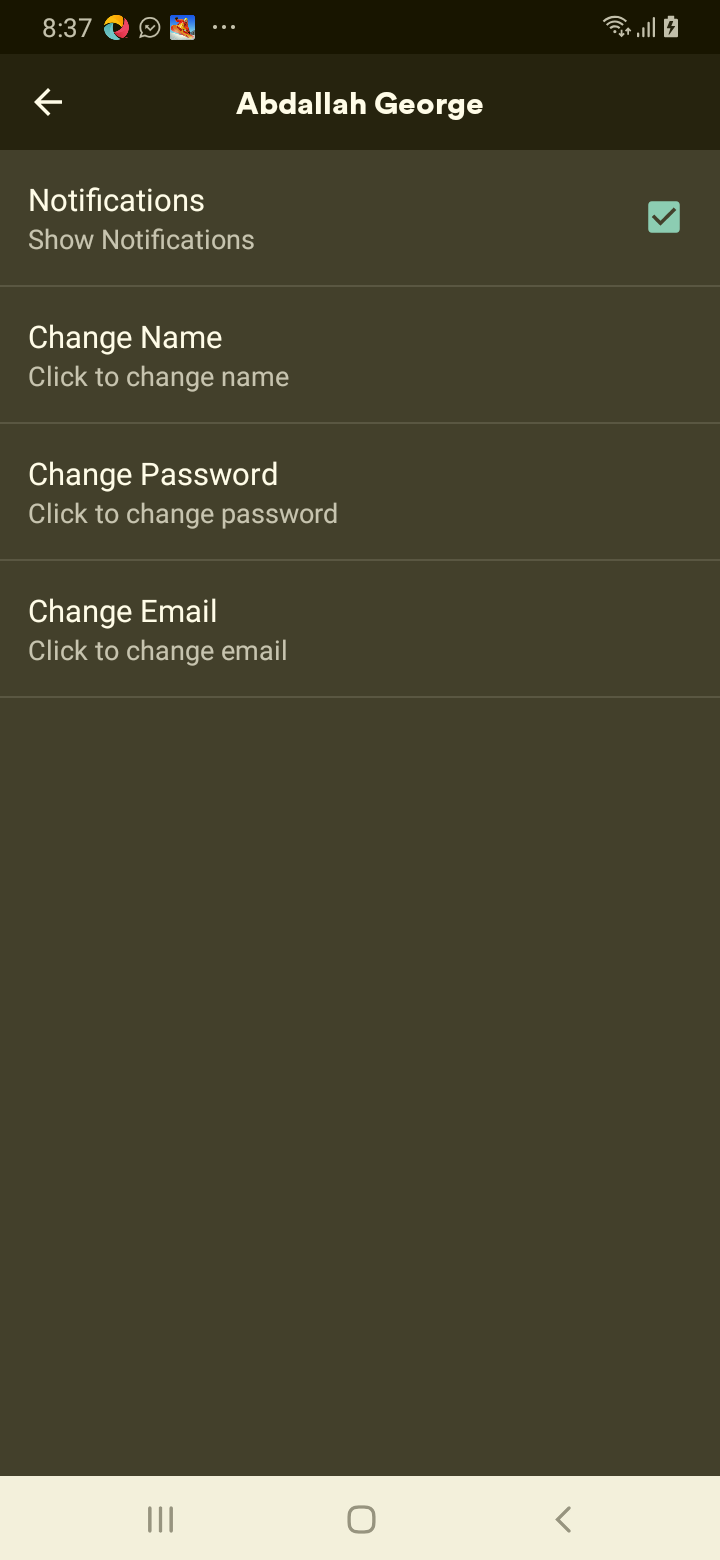
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d185670>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user name to empty name")  @allure.title("changes the user name to empty name")  @allure.description("changes the user name to empty name")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test8  def test\_case\_8(self, setup):  """  changes the user name to empty name  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user name to empty name")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_name("", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_name("Abdallah George", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test changes the user name to empty name") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:375: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test changes the user name to empty name



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user name to name + characters name - failed

changes the user name to name + characters name

## Details

AssertionError: assert False

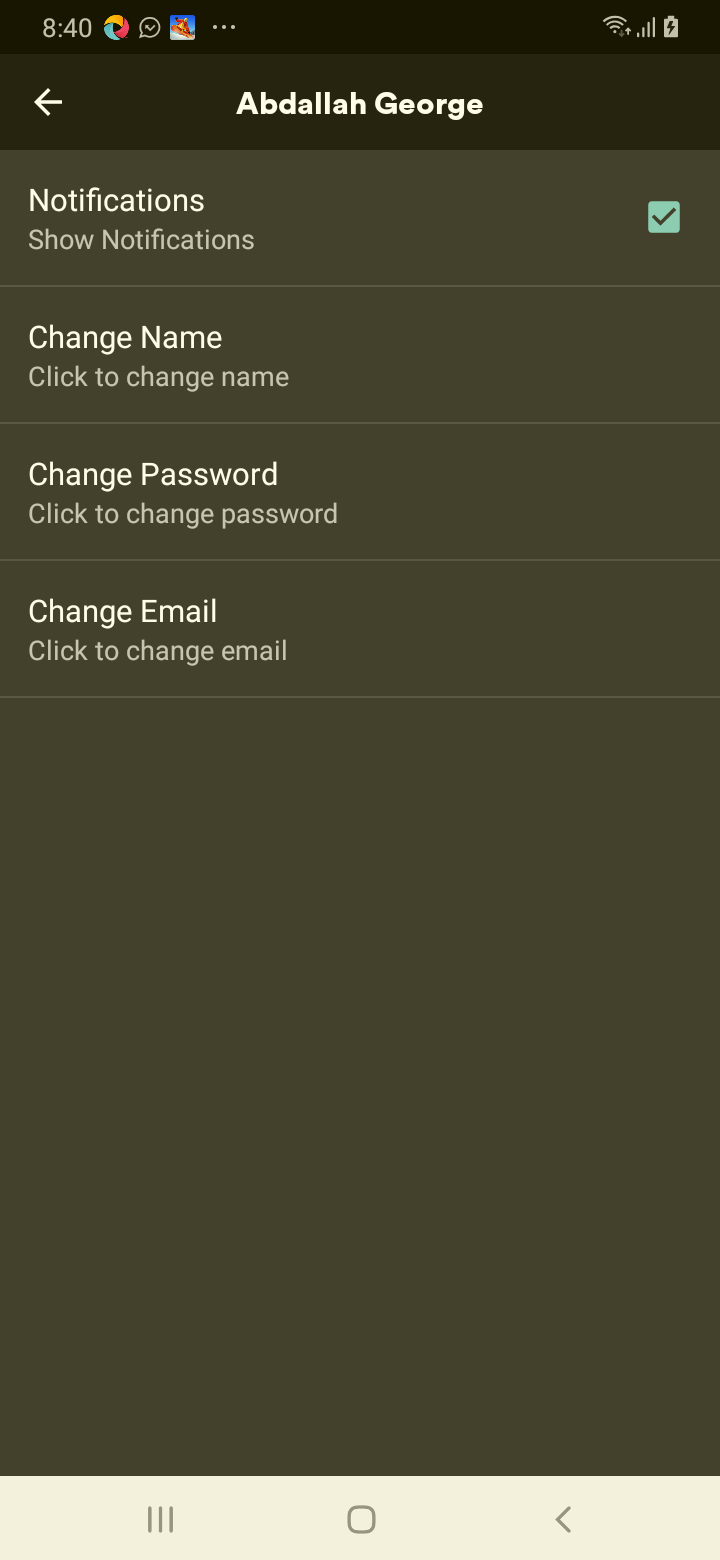
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d18b8e0>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user name to name + characters name")  @allure.title("changes the user name to name + characters name")  @allure.description("changes the user name to name + characters name")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test12  def test\_case\_12(self, setup):  """  changes the user name to name + characters name  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user name to name + characters name")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_name("kamel12345\_!@#!@$", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_name("Abdallah George", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test changes the user name to name + characters name") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:523: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test changes the user name to name + characters name



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user name to numbers name - failed

changes the user name to numbers name

## Details

AssertionError: assert False

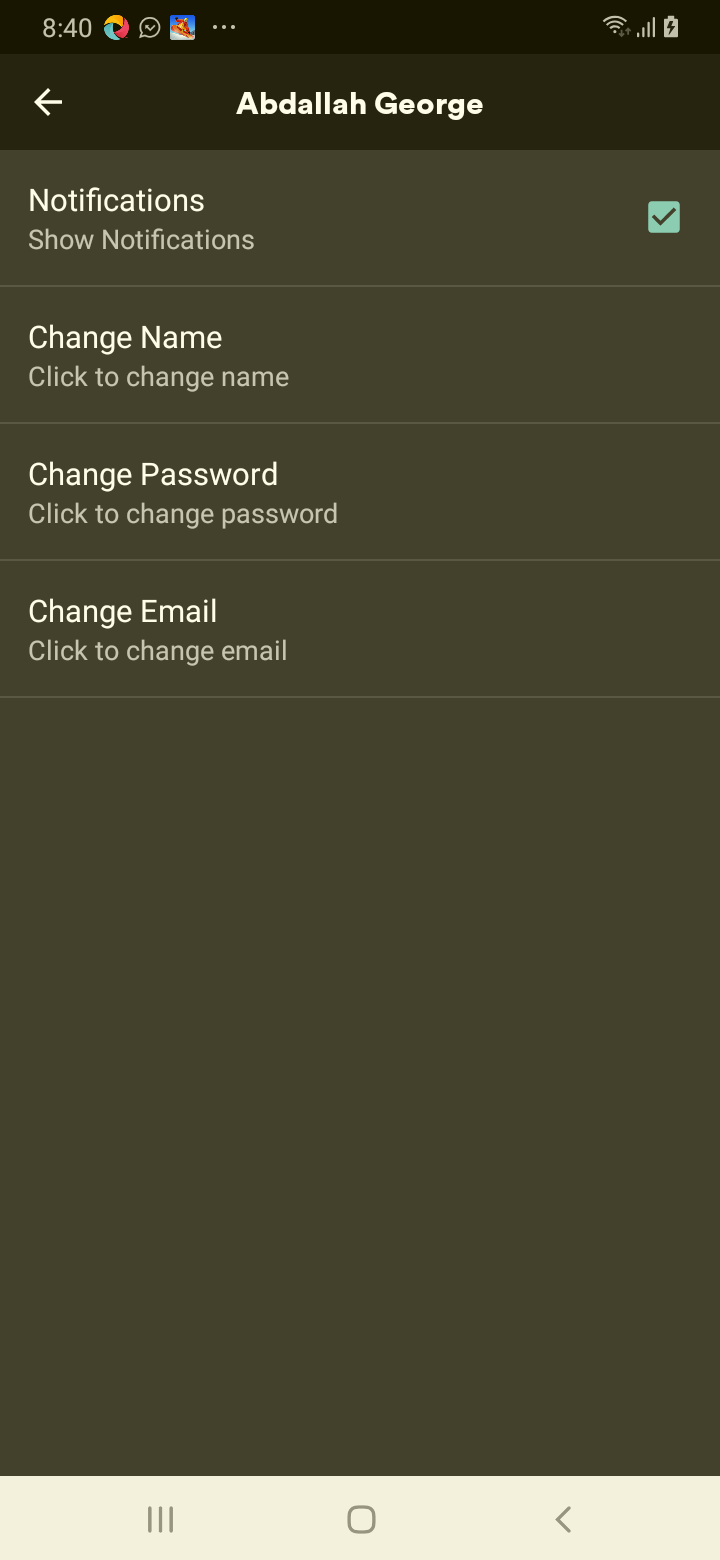
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d18f370>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user name to numbers name")  @allure.title("changes the user name to numbers name")  @allure.description("changes the user name to numbers name")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test11  def test\_case\_11(self, setup):  """  changes the user name to numbers name  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user name to numbers name")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_name("12345", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_name("Abdallah George", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test changes the user name to numbers name") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:486: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test changes the user name to numbers name



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user name to spaces name - failed

changes the user name to spaces name

## Details

AssertionError: assert False

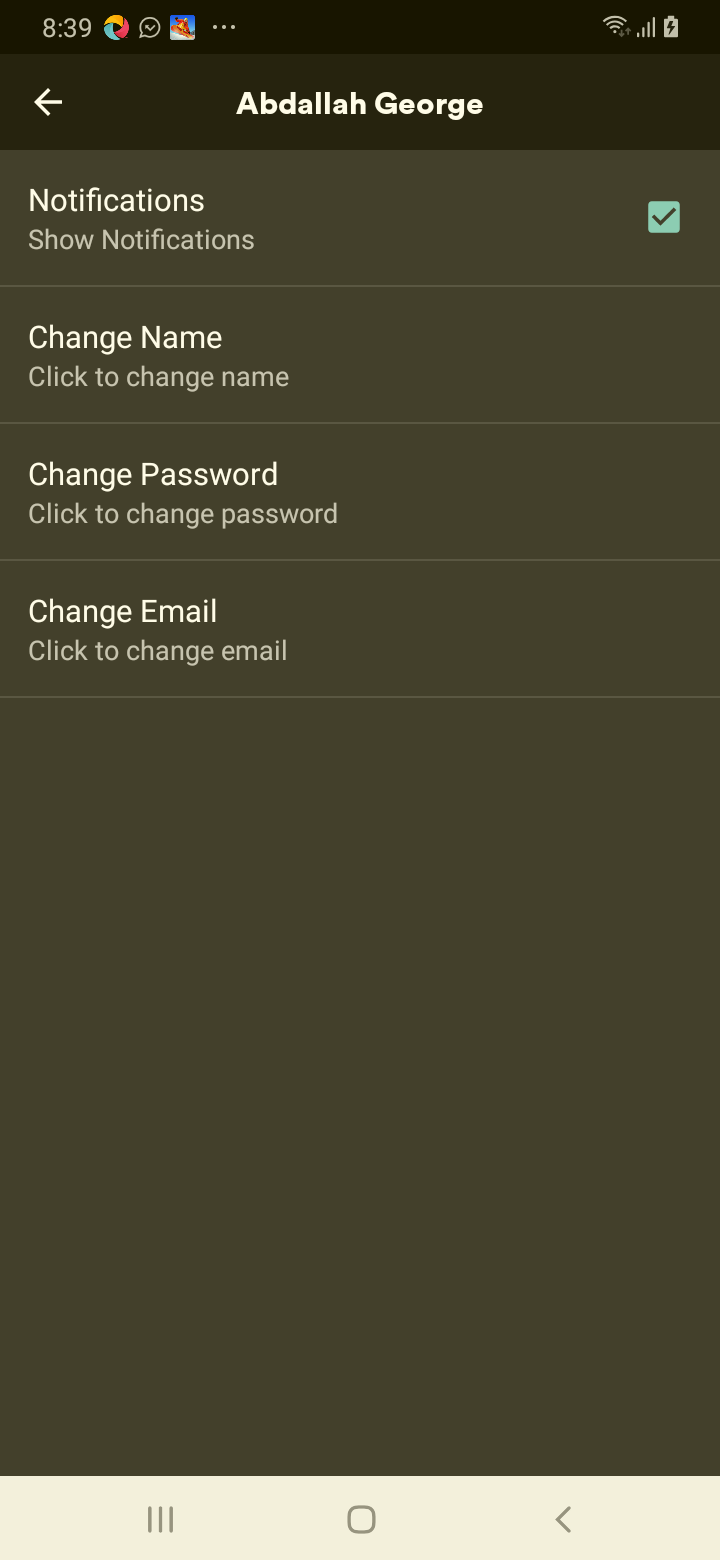
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d18bd60>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user name to spaces name")  @allure.title("changes the user name to spaces name")  @allure.description("changes the user name to spaces name")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test10  def test\_case\_10(self, setup):  """  changes the user name to spaces name  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user name to spaces name")  assert False  pp = ProfilePage(self.driver)  pp.check\_edit\_profile\_button()    if not pp.change\_name(" ", "or"):  Helper.report\_allure(self.driver, "Passed")  assert True  else:  pp.change\_name("Abdallah George", "re")  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test changes the user name to spaces name") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:449: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test changes the user name to spaces name



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user password to !@!#@ password - failed

changes the user password to !@!#@ password

## Details

AssertionError: assert False

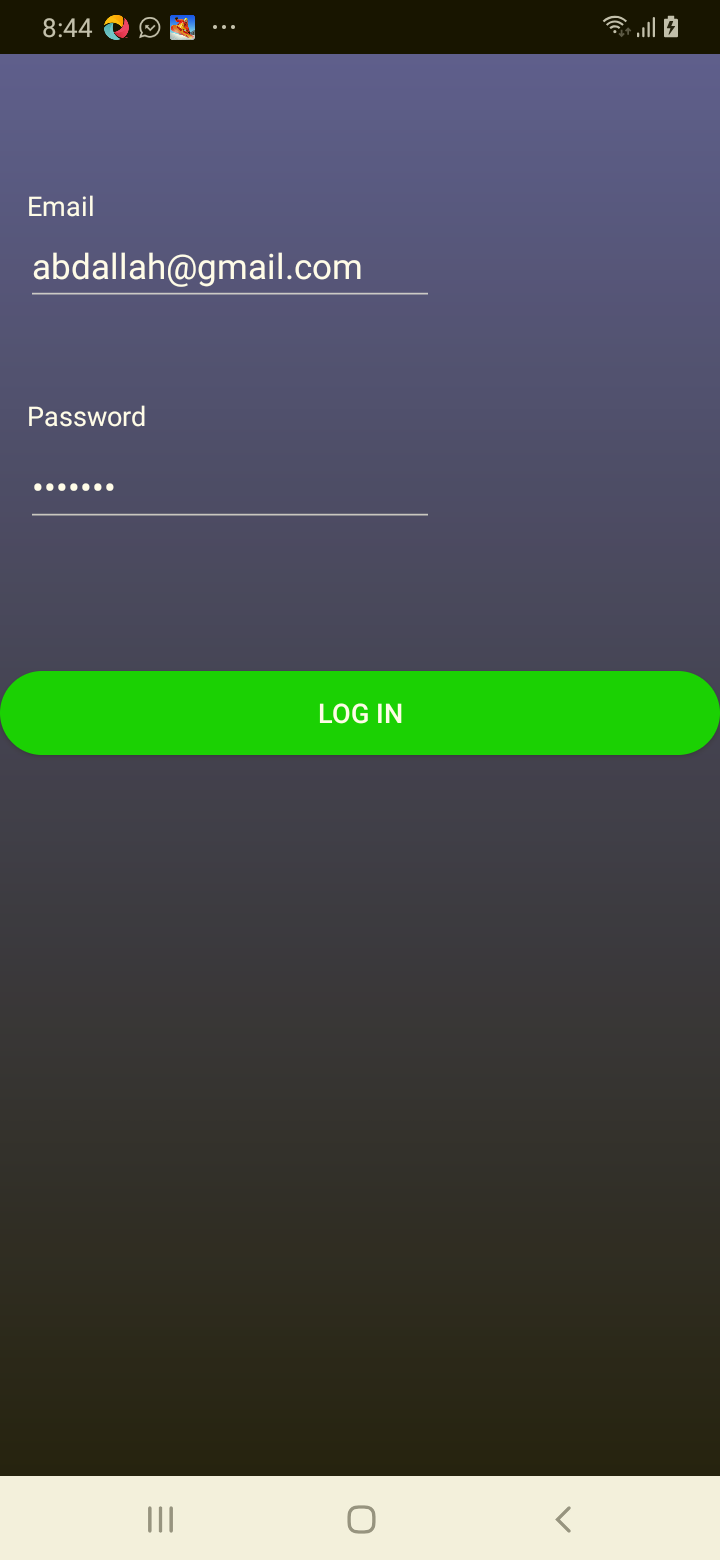
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d185790>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user password to !@!#@ password")  @allure.title("changes the user password to !@!#@ password")  @allure.description("changes the user password to !@!#@ password")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test15  def test\_case\_15(self, setup):  """  changes the user password to !@!#@ password  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user password to !@!#@ password") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:622: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user password to !@!#@ password



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user password to empty password - failed

changes the user password to empty password

## Details

AssertionError: assert False

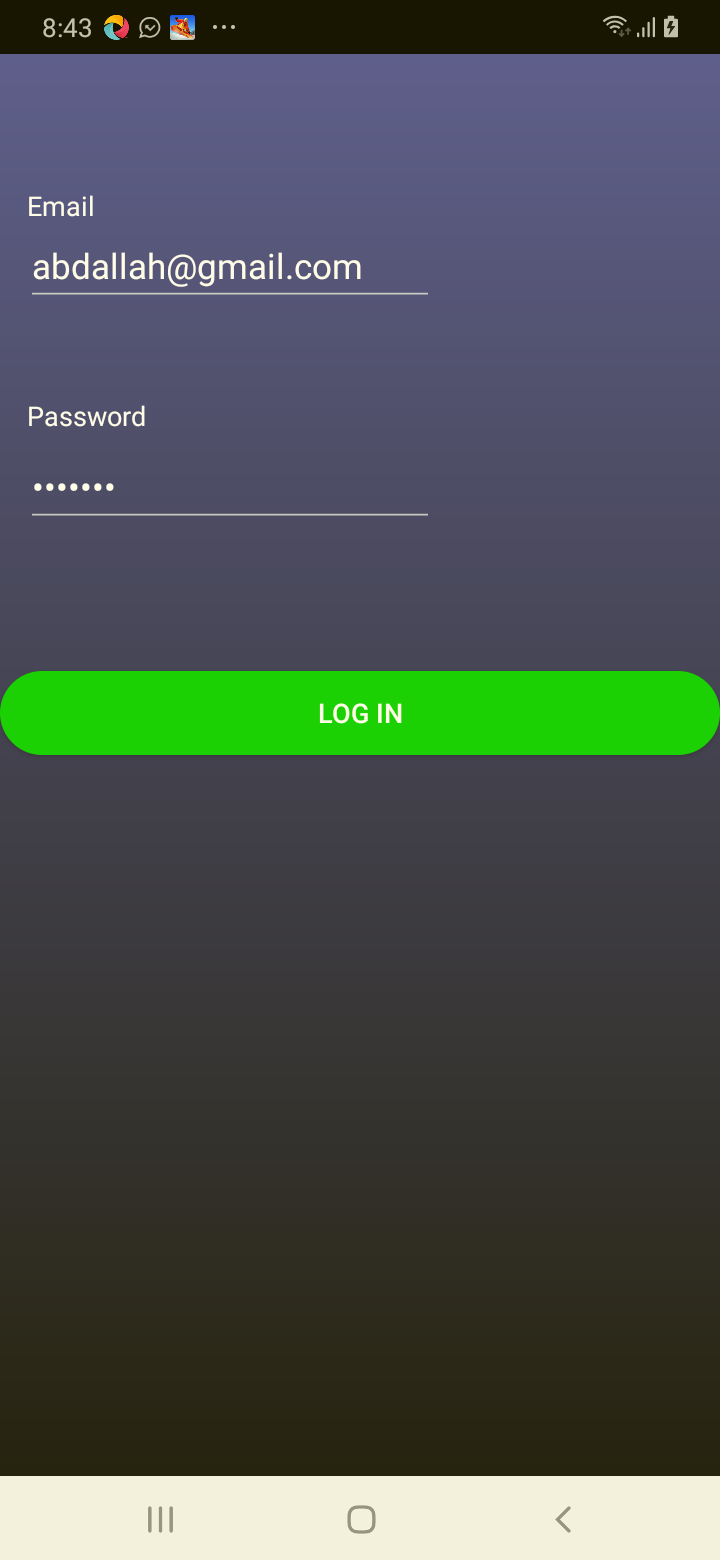
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d349940>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user password to empty password")  @allure.title("changes the user password to empty password")  @allure.description("changes the user password to empty password")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test14  def test\_case\_14(self, setup):  """  changes the user password to empty password  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user password to empty password") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:585: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user password to empty password



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# changes the user password to spaces password - failed

changes the user password to spaces password

## Details

AssertionError: assert False

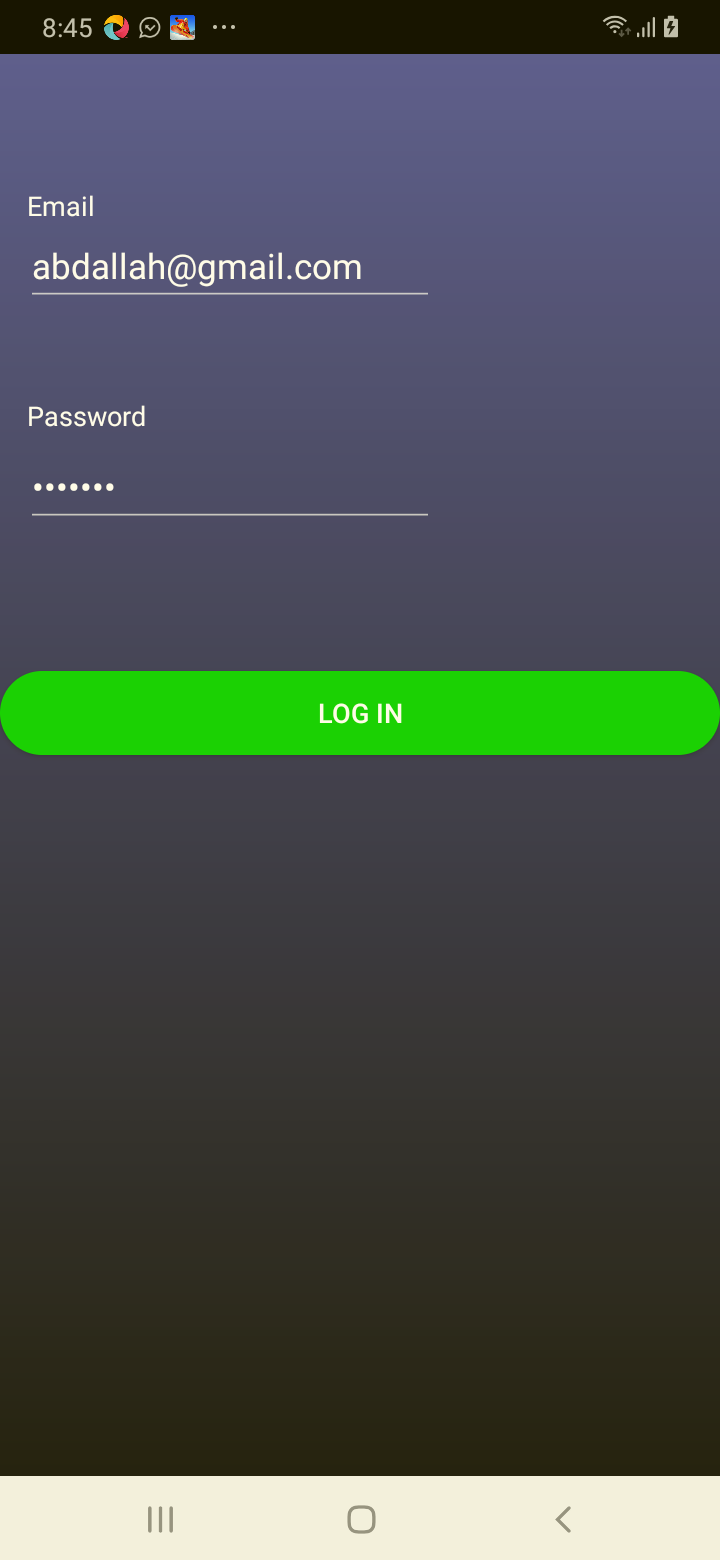
|  |
| --- |
| self = <test\_profile.TestProfile object at 0x10d2ece80>, setup = None   @allure.severity(allure.severity\_level.CRITICAL)  @allure.story("Profile Tests")  @allure.sub\_suite("changes the user password to spaces password")  @allure.title("changes the user password to spaces password")  @allure.description("changes the user password to spaces password")  @pytest.mark.Do  @pytest.mark.Profile  @pytest.mark.Test16  def test\_case\_16(self, setup):  """  changes the user password to spaces password  """  ap = AuthenticationPage(self.driver)  ap.click\_signin\_button()  lp = LoginPage(self.driver)  lp.do\_the\_login(Constants.correct\_credentials["email"], Constants.correct\_credentials["password"])  hp = HomePage(self.driver)  hp.click\_premium\_button()    pp = PremiumPage(self.driver)  if not pp.enter\_profile():  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "failed to test change the user password to spaces password") > assert False E assert False  Mobile\_Testing/Tests/test\_profile.py:659: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] failed to test change the user password to spaces password



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Checking delete track - passed

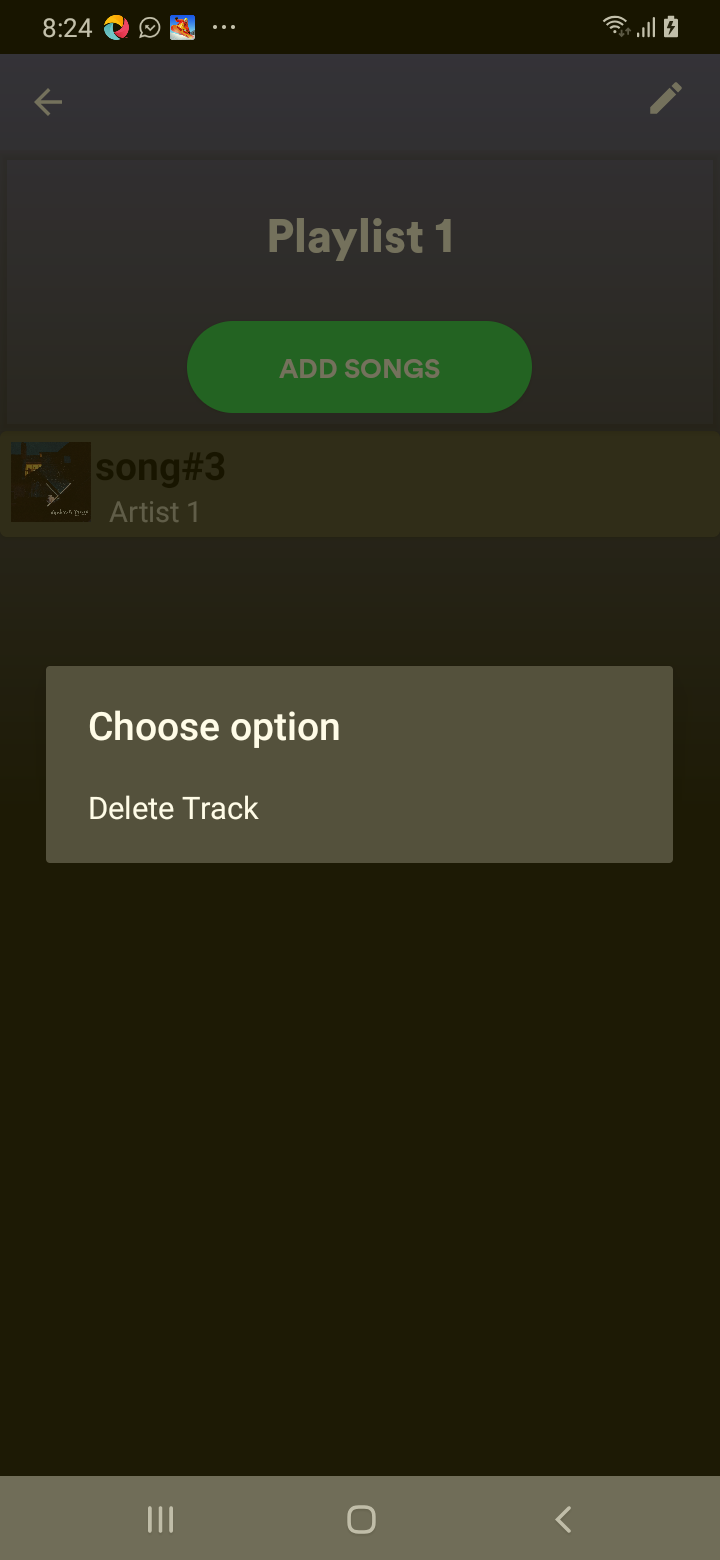
Checking delete track exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] delete track exists



## Test Teardown

[Fixture] setup::0

# Checking rename - passed

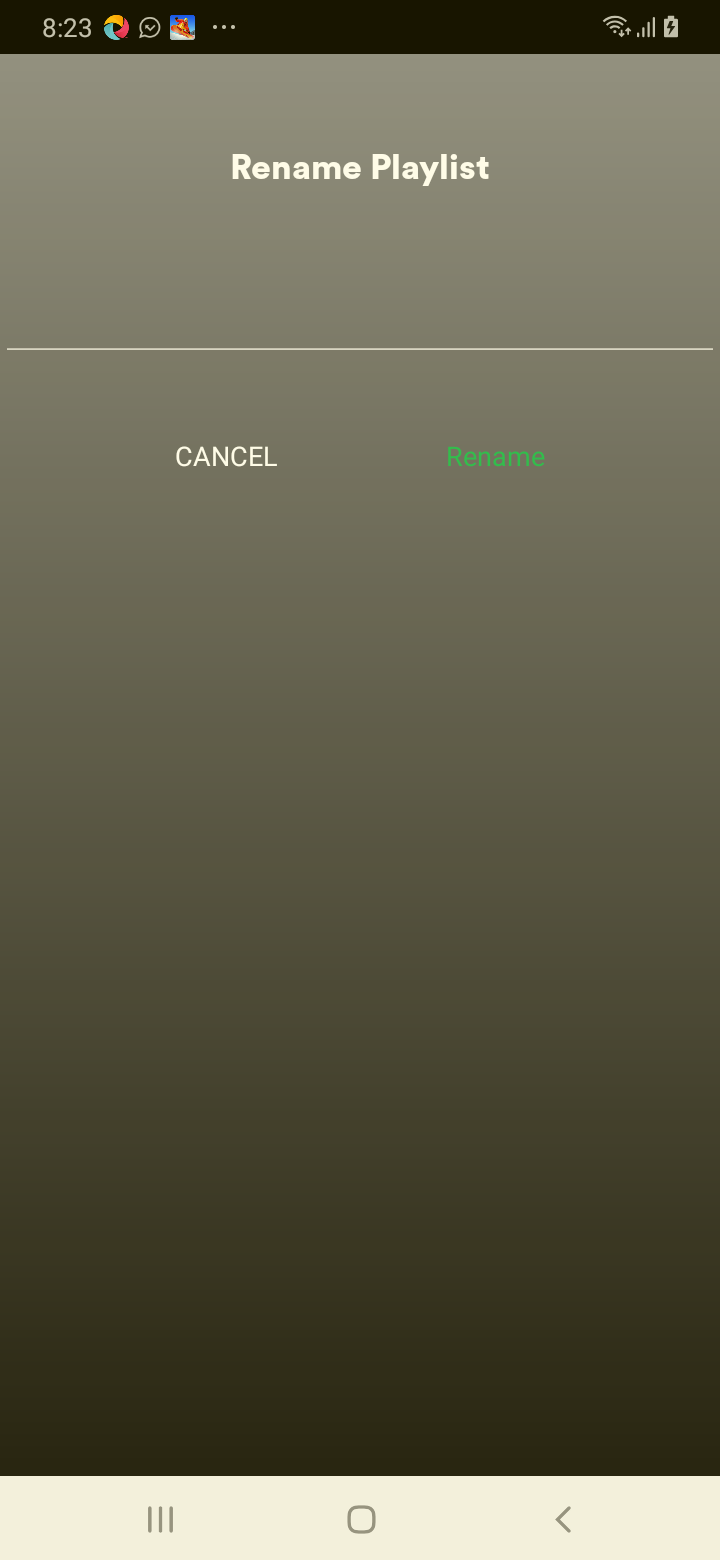
Checking rename exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] rename exists



## Test Teardown

[Fixture] setup::0

# Checking ad breaks | ad free music box - passed

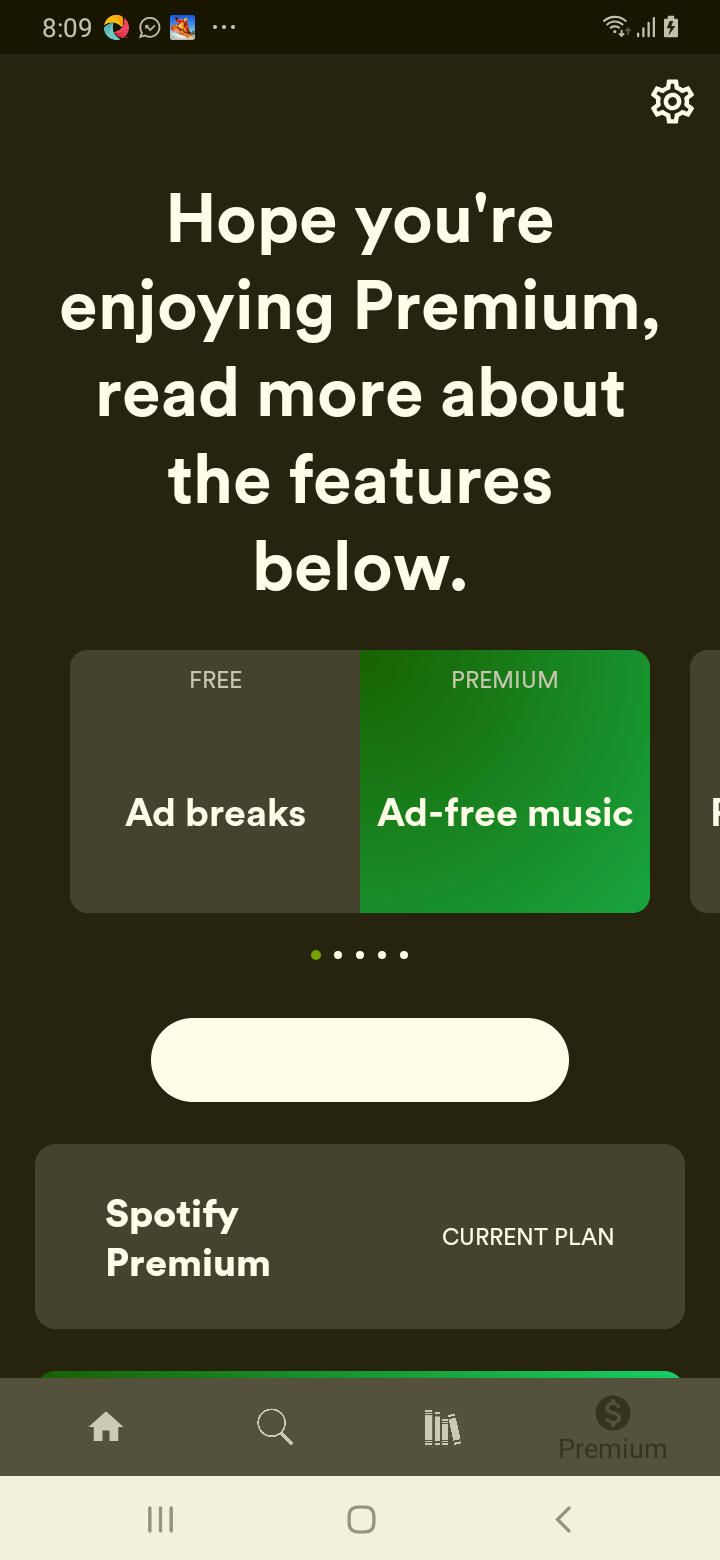
Checking ad breaks | ad free music box exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] break box exists



## Test Teardown

[Fixture] setup::0

# Checking cancel button - passed

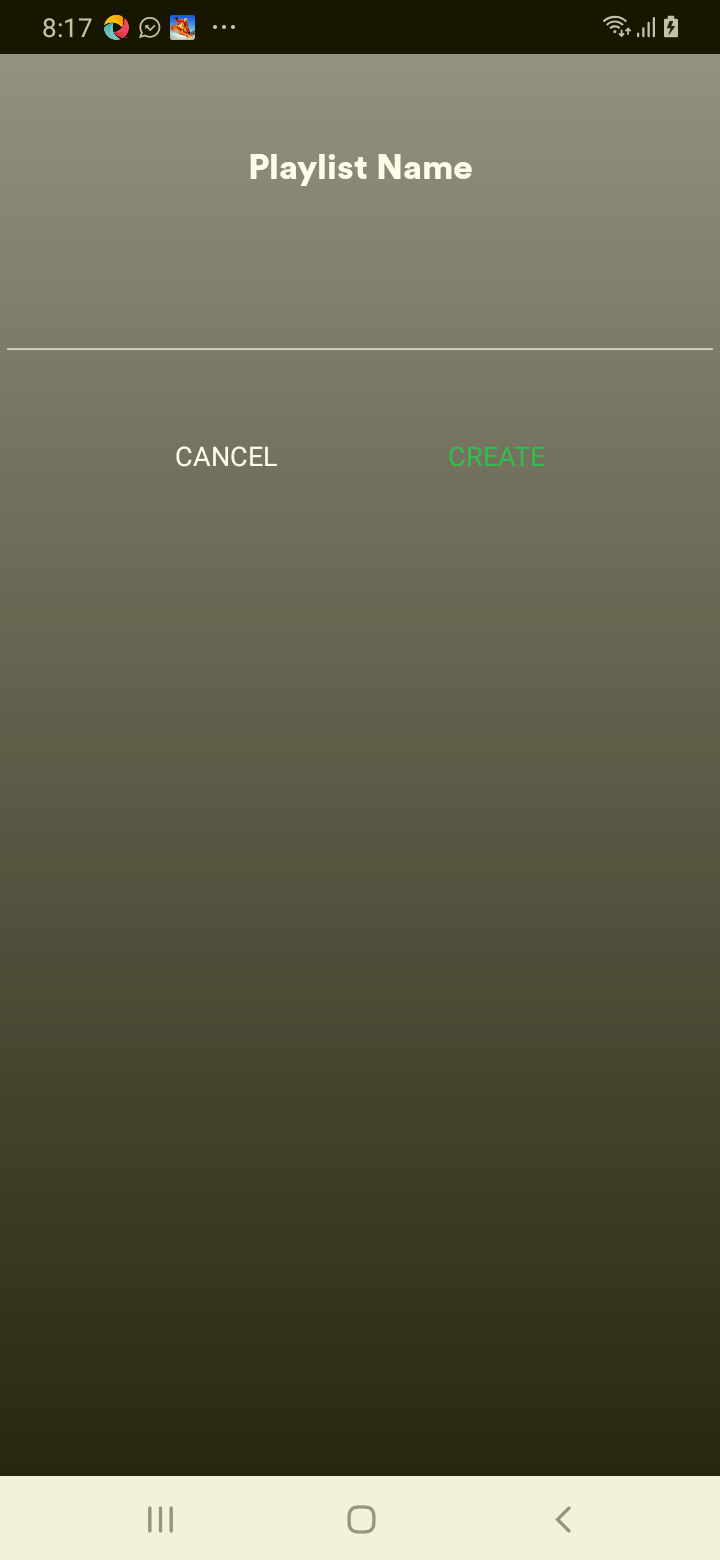
Checking cancel button exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] cancel button exist



## Test Teardown

[Fixture] setup::0

# Checking cancel rename - passed

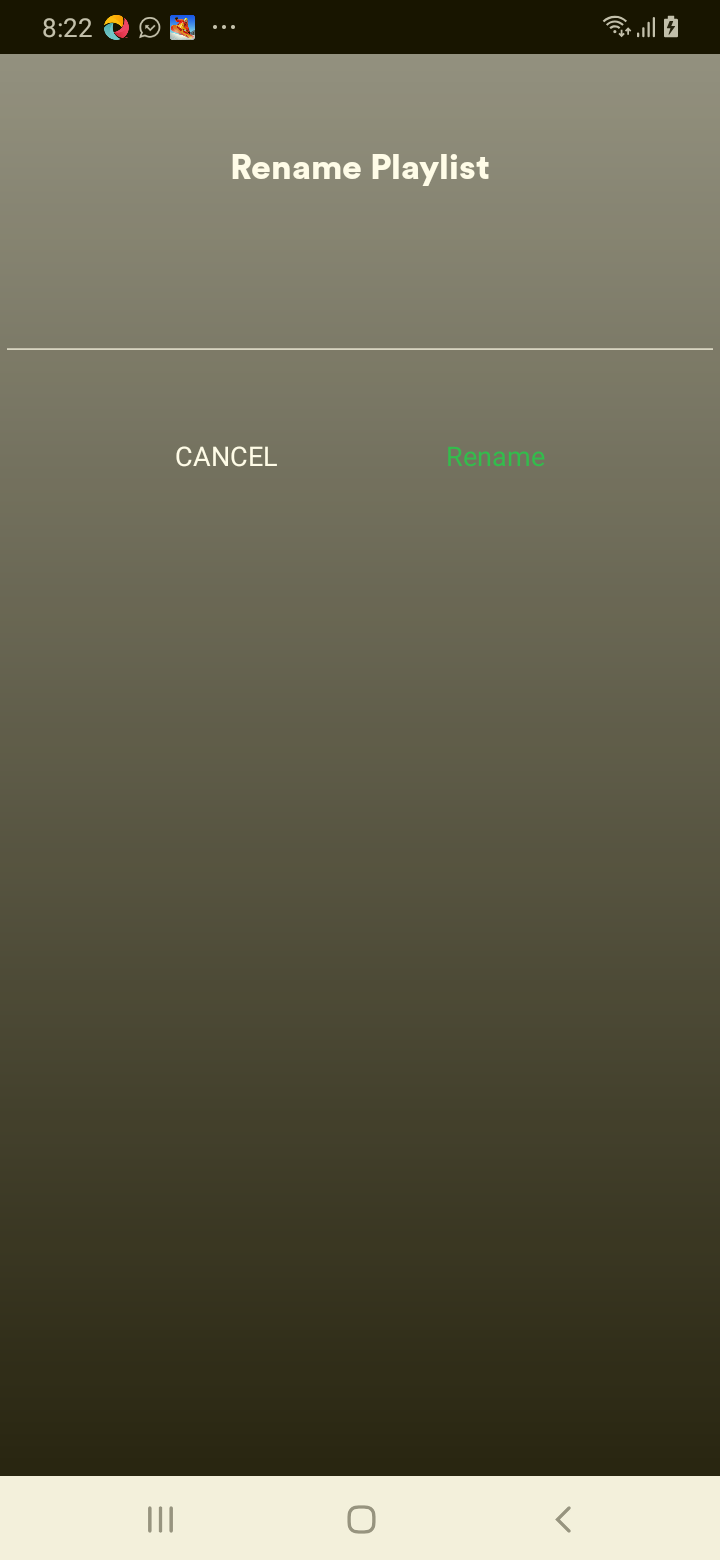
Checking cancel rename exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] cancel rename exists



## Test Teardown

[Fixture] setup::0

# Checking canceling premium - passed

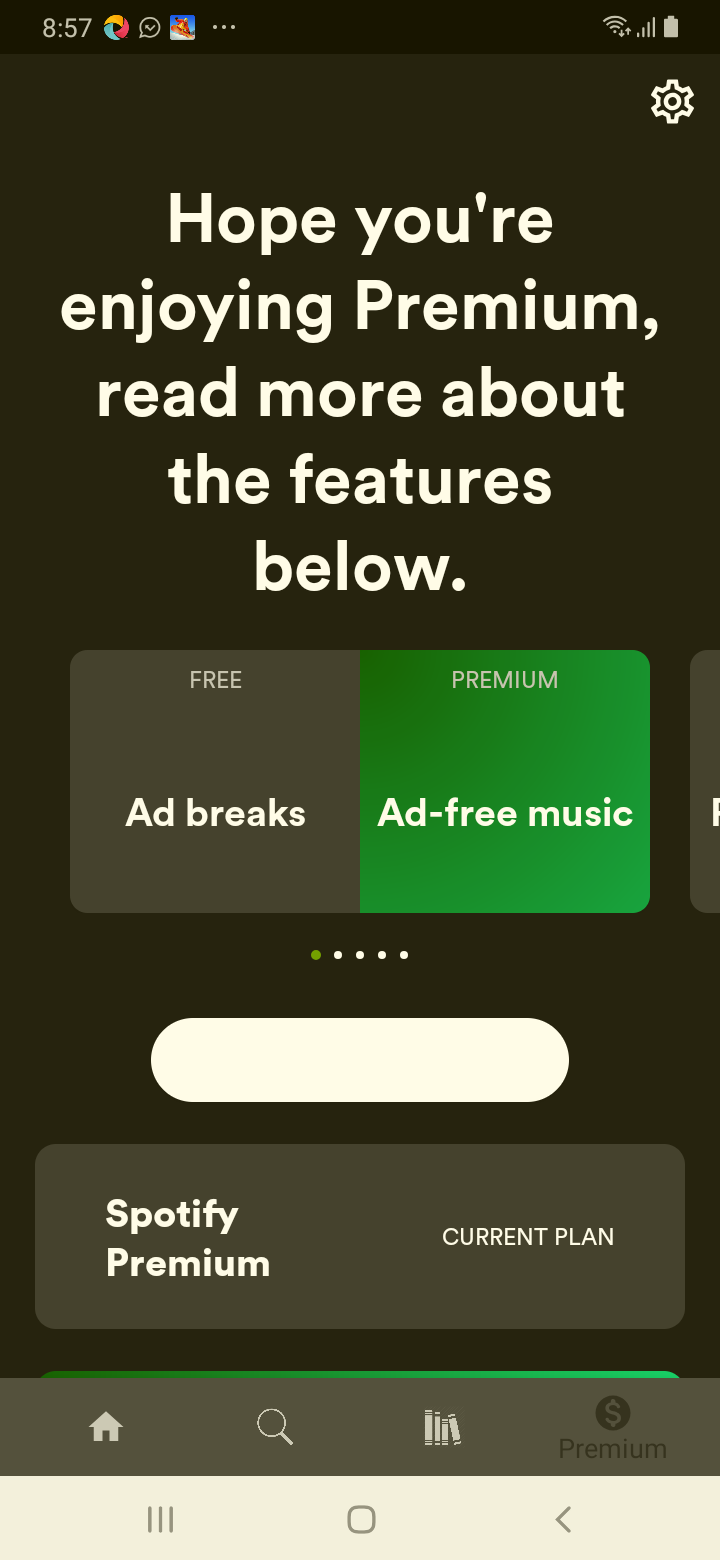
Checking canceling premium

## Test Setup

[Fixture] setup

## Test Body

[Attachment] premium canceled



## Test Teardown

[Fixture] setup::0

# Checking create playlist - passed

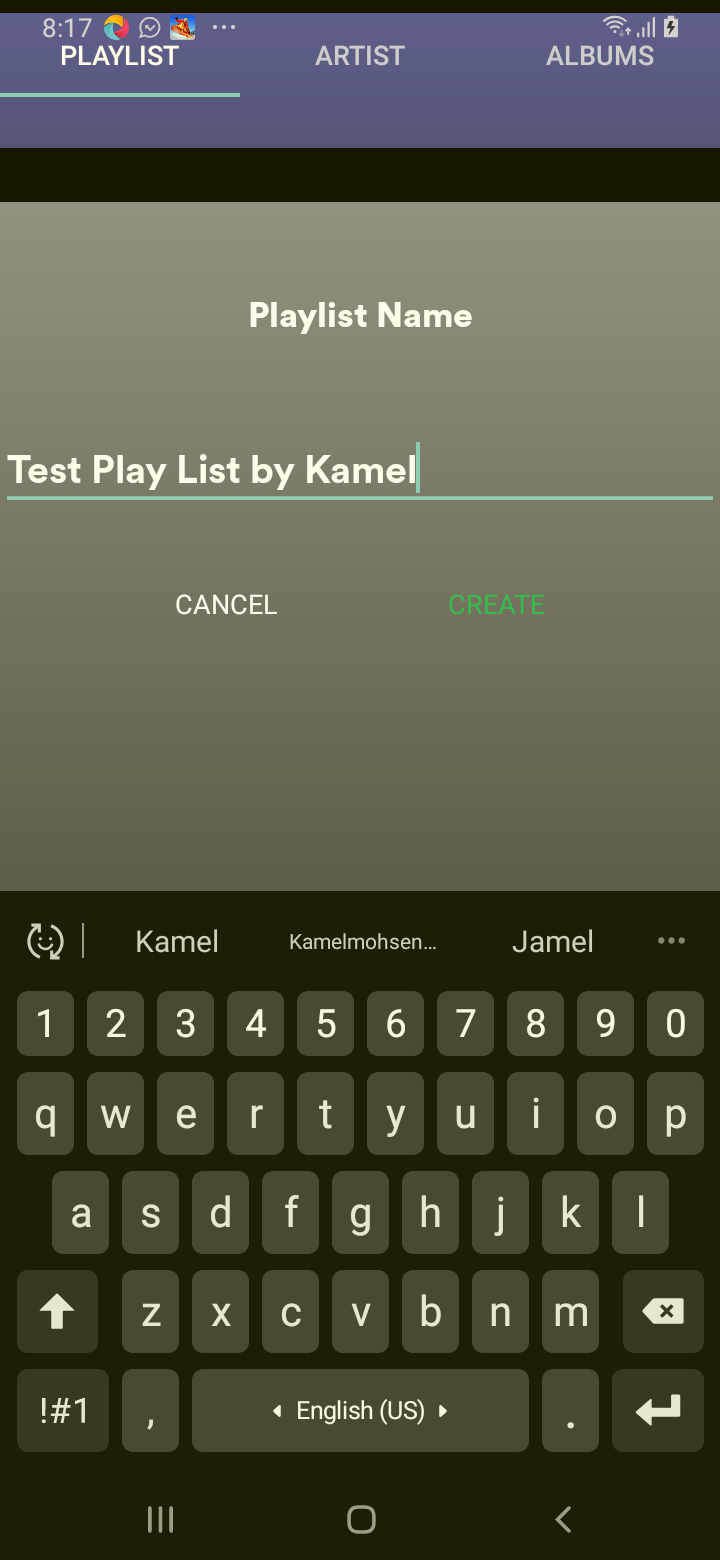
Checking create playlist

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Creation done



## Test Teardown

[Fixture] setup::0

# Checking create playlist button - passed

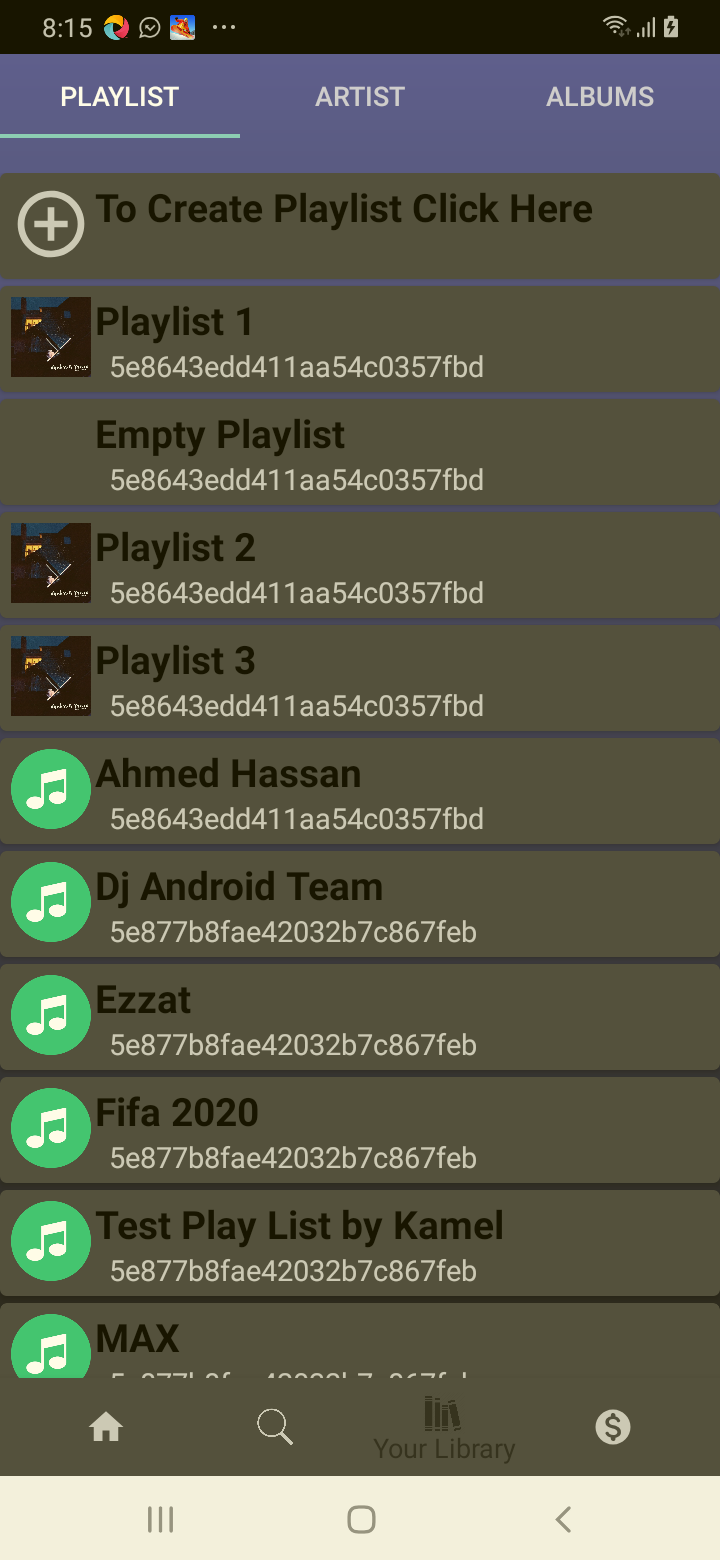
Checking create playlist button exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] create button exist



## Test Teardown

[Fixture] setup::0

# Checking create playlist with '@!##!@' name - passed

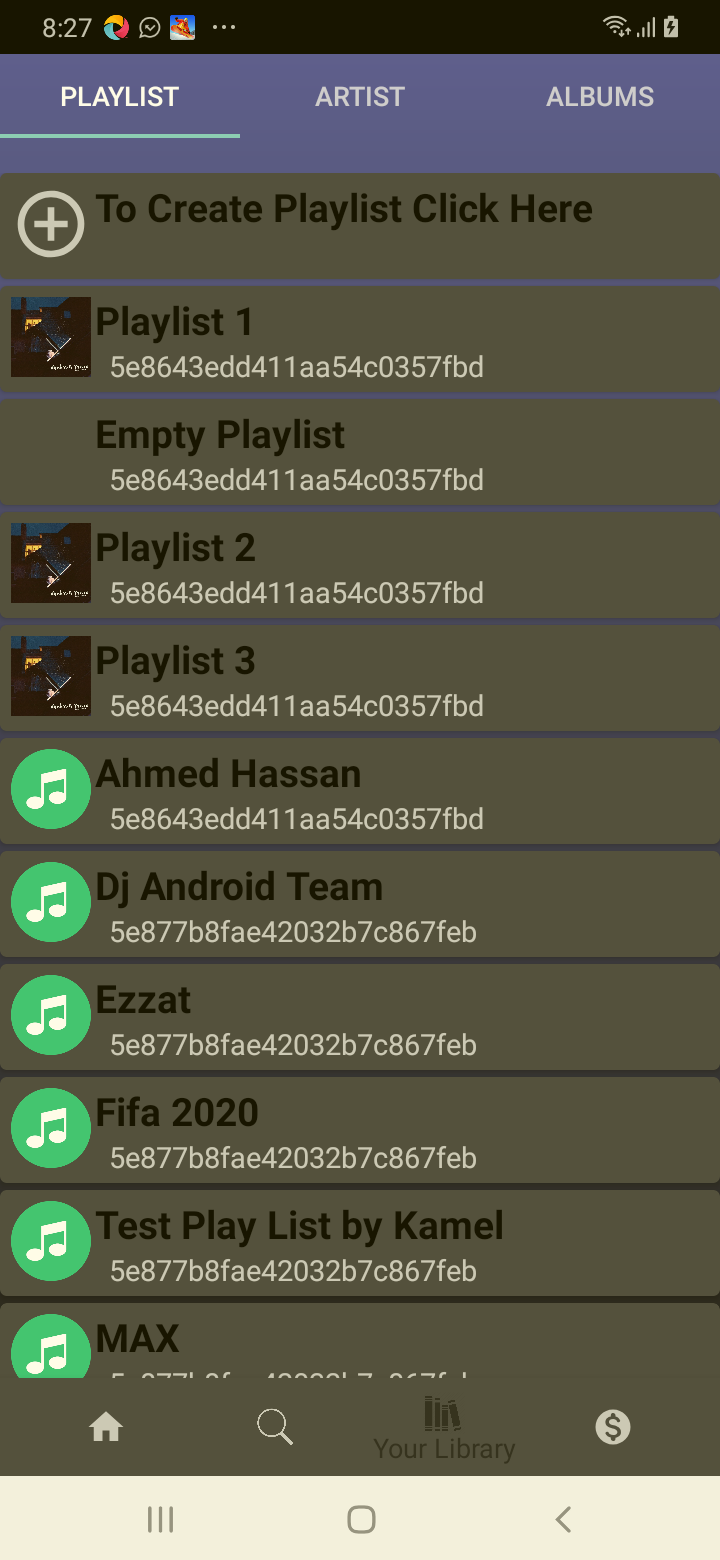
Checking create playlist with '@!##!@' name

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Creation failed



## Test Teardown

[Fixture] setup::0

# Checking create playlist with spaces name - passed

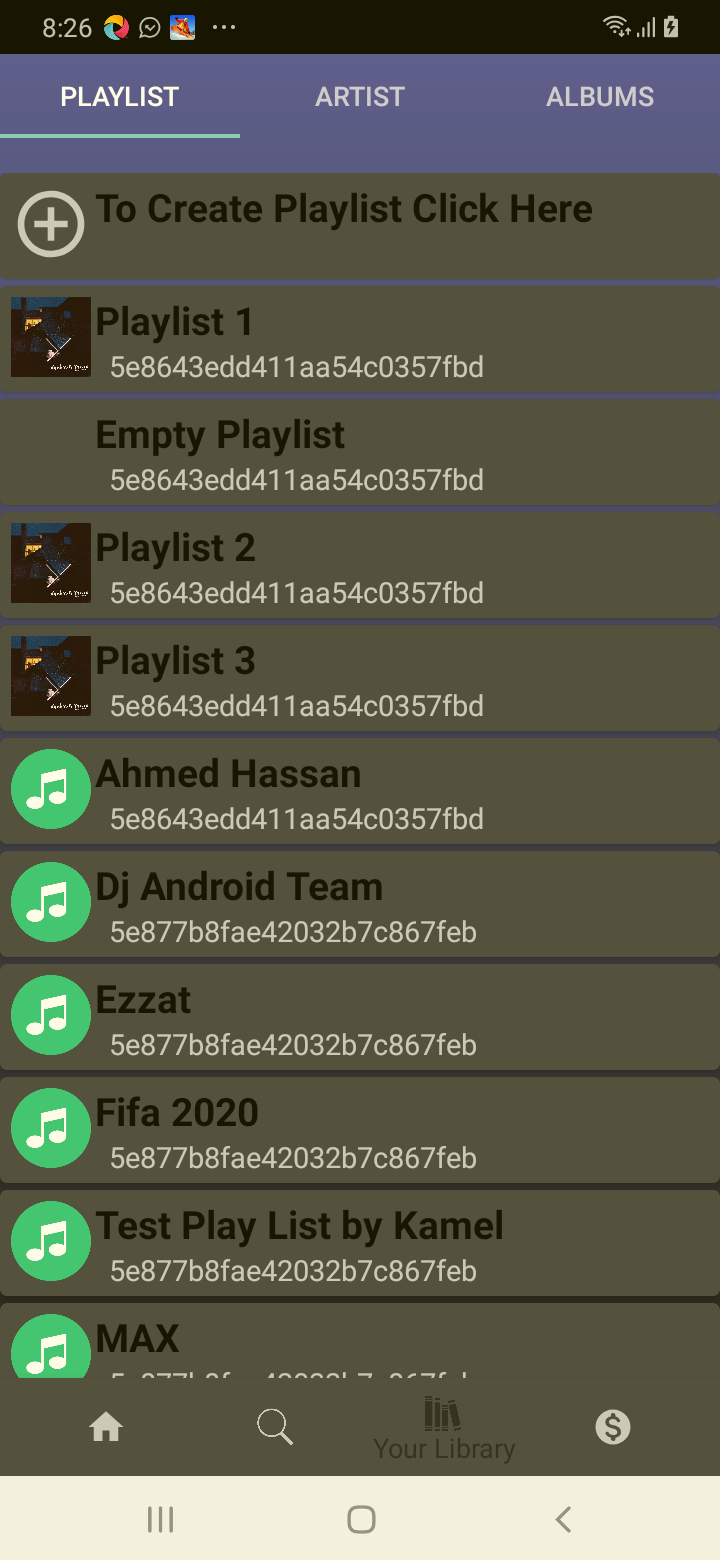
Checking create playlist with spaces name

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Creation failed



## Test Teardown

[Fixture] setup::0

# Checking deleting playlist - passed

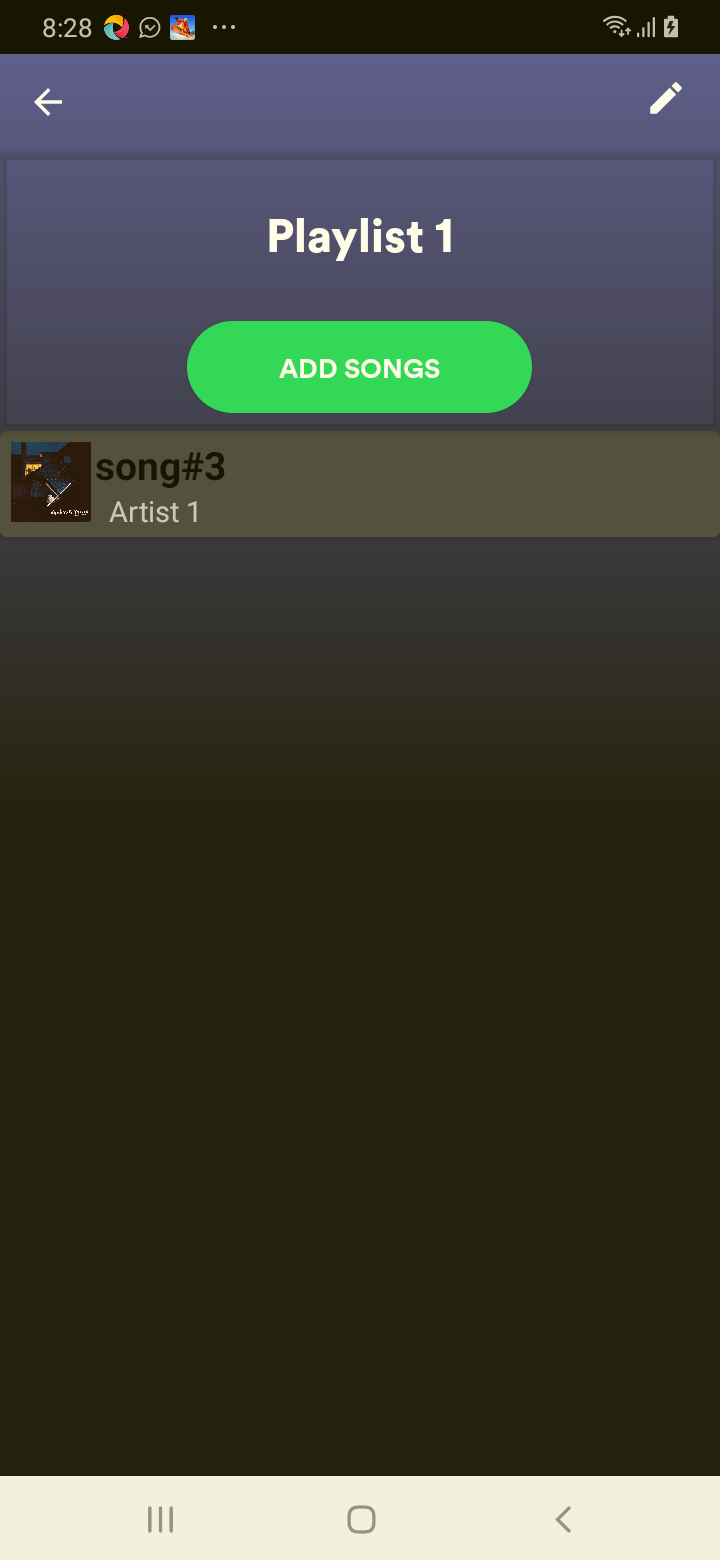
Checking deleting playlist

## Test Setup

[Fixture] setup

## Test Body

[Attachment] deletion success



## Test Teardown

[Fixture] setup::0

# Checking first album - passed

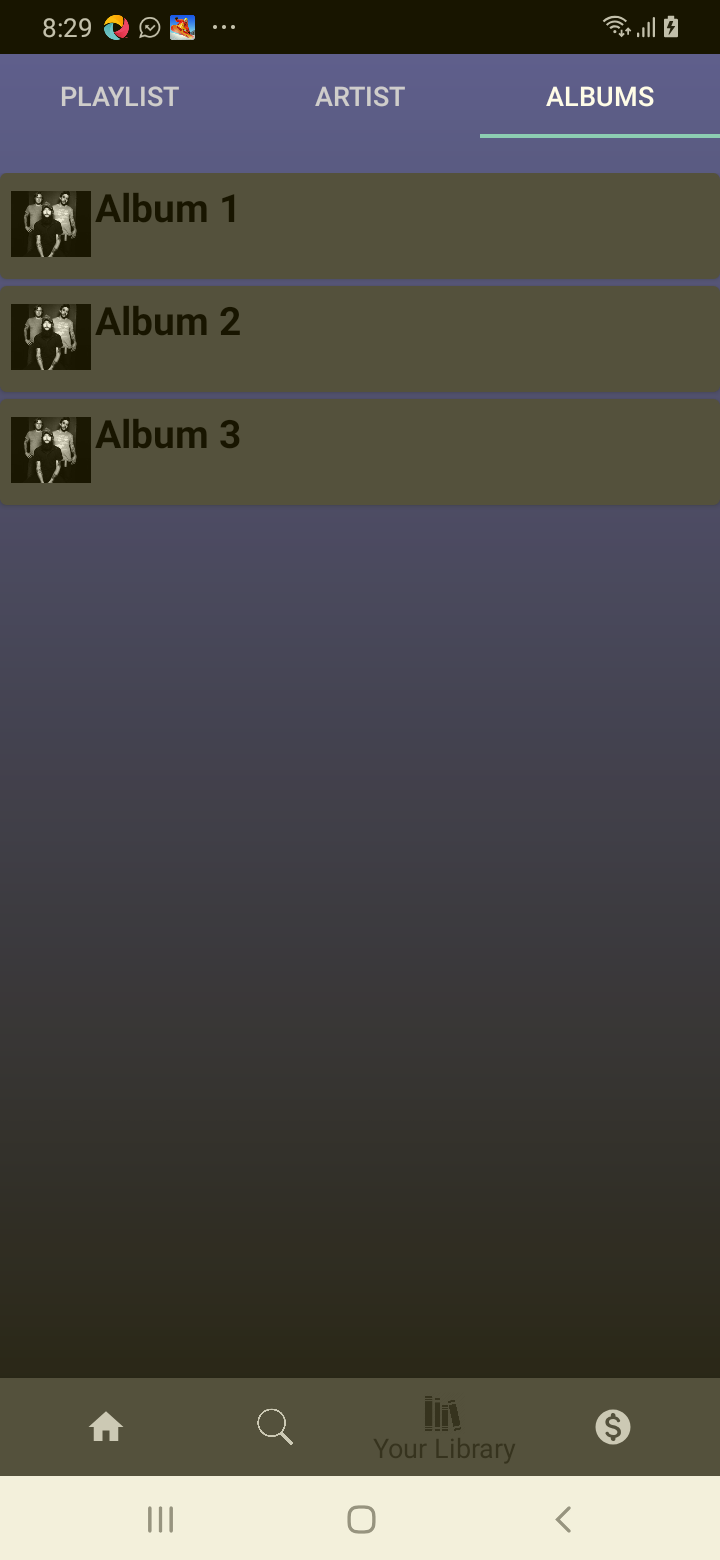
Checking first album exists and is click-able

## Test Setup

[Fixture] setup

## Test Body

[Attachment] first album exists



## Test Teardown

[Fixture] setup::0

# Checking first song - passed

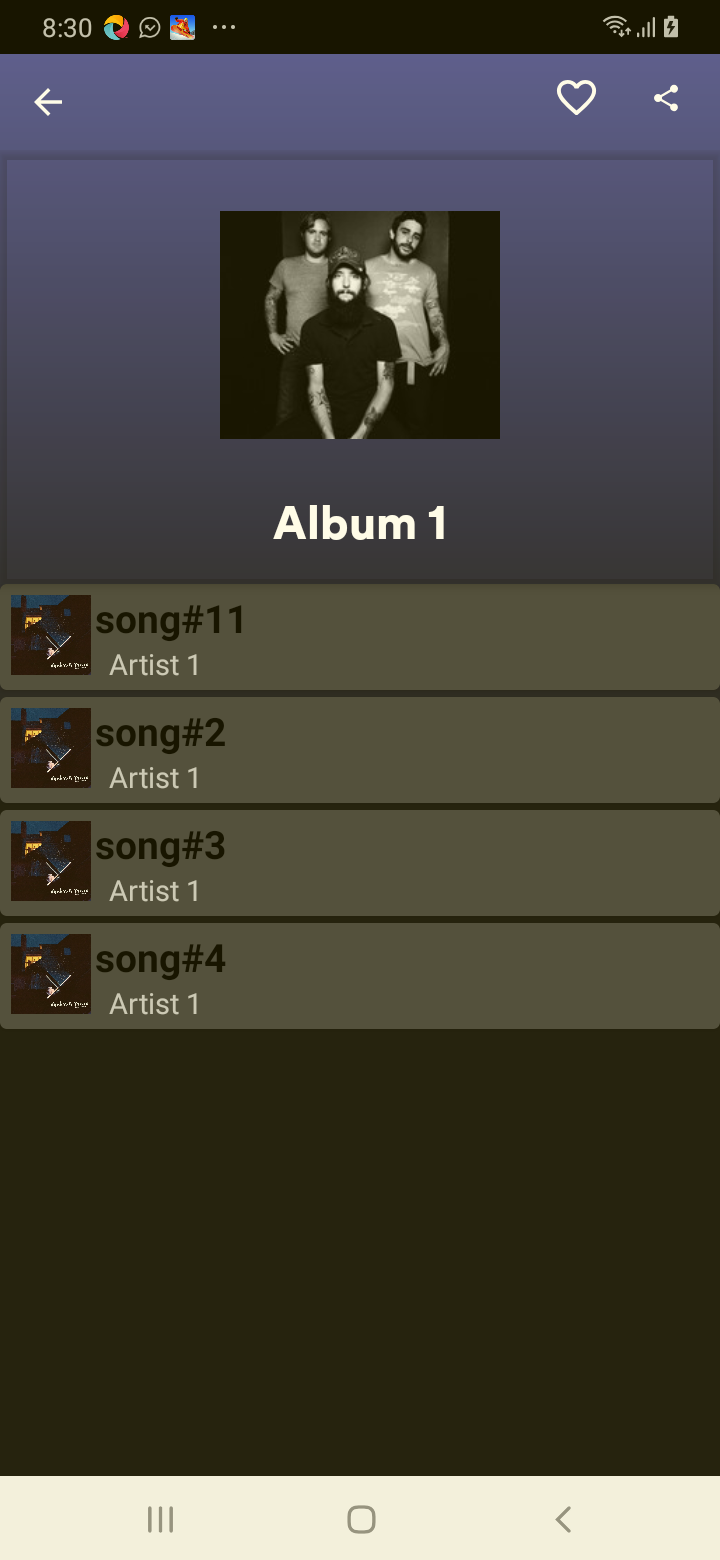
Checking first song exists and is click-able

## Test Setup

[Fixture] setup

## Test Body

[Attachment] first song exists



## Test Teardown

[Fixture] setup::0

# Checking first song in edit playlist - passed

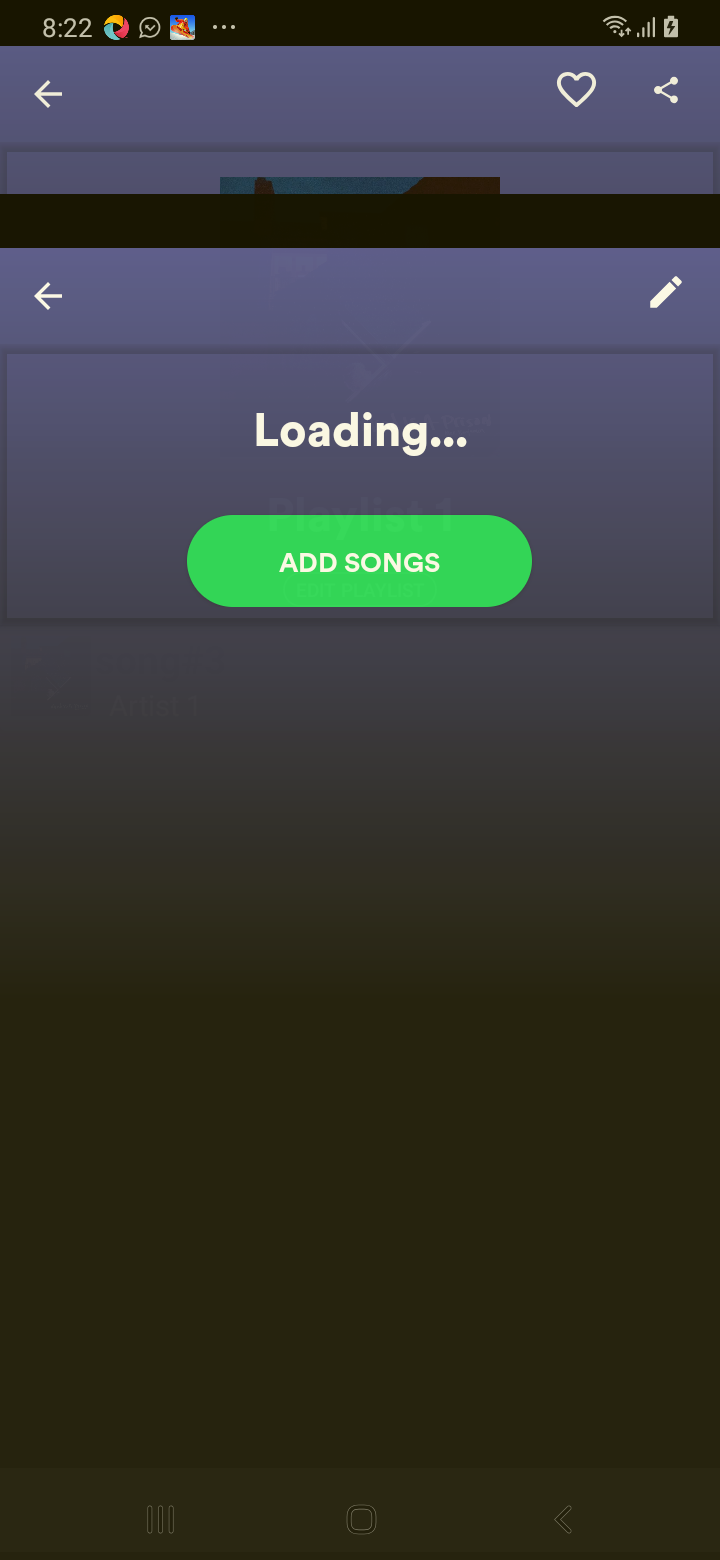
Checking first song in edit exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] first song in edit exists



## Test Teardown

[Fixture] setup::0

# Checking free box - passed

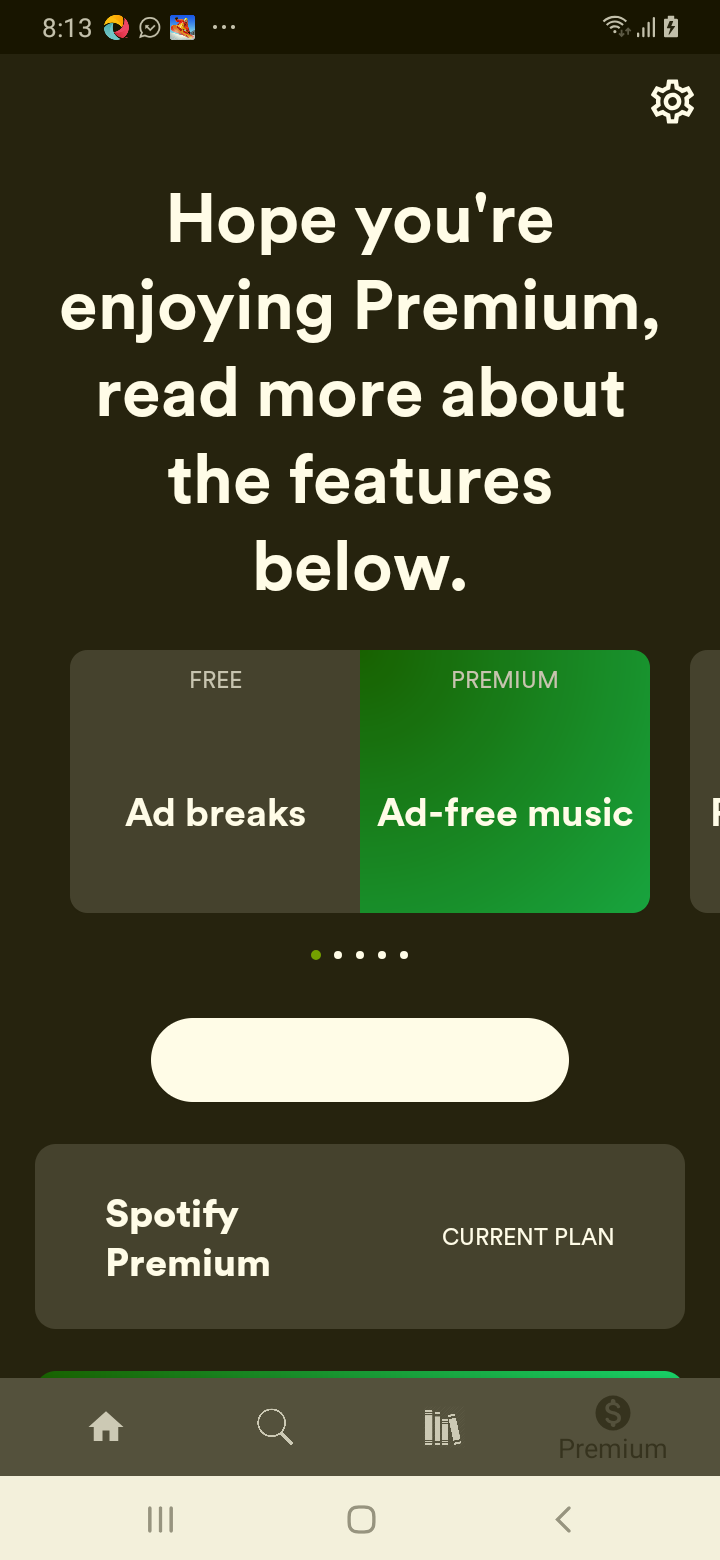
Checking free box exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] free box exists



## Test Teardown

[Fixture] setup::0

# Checking get premium button - passed

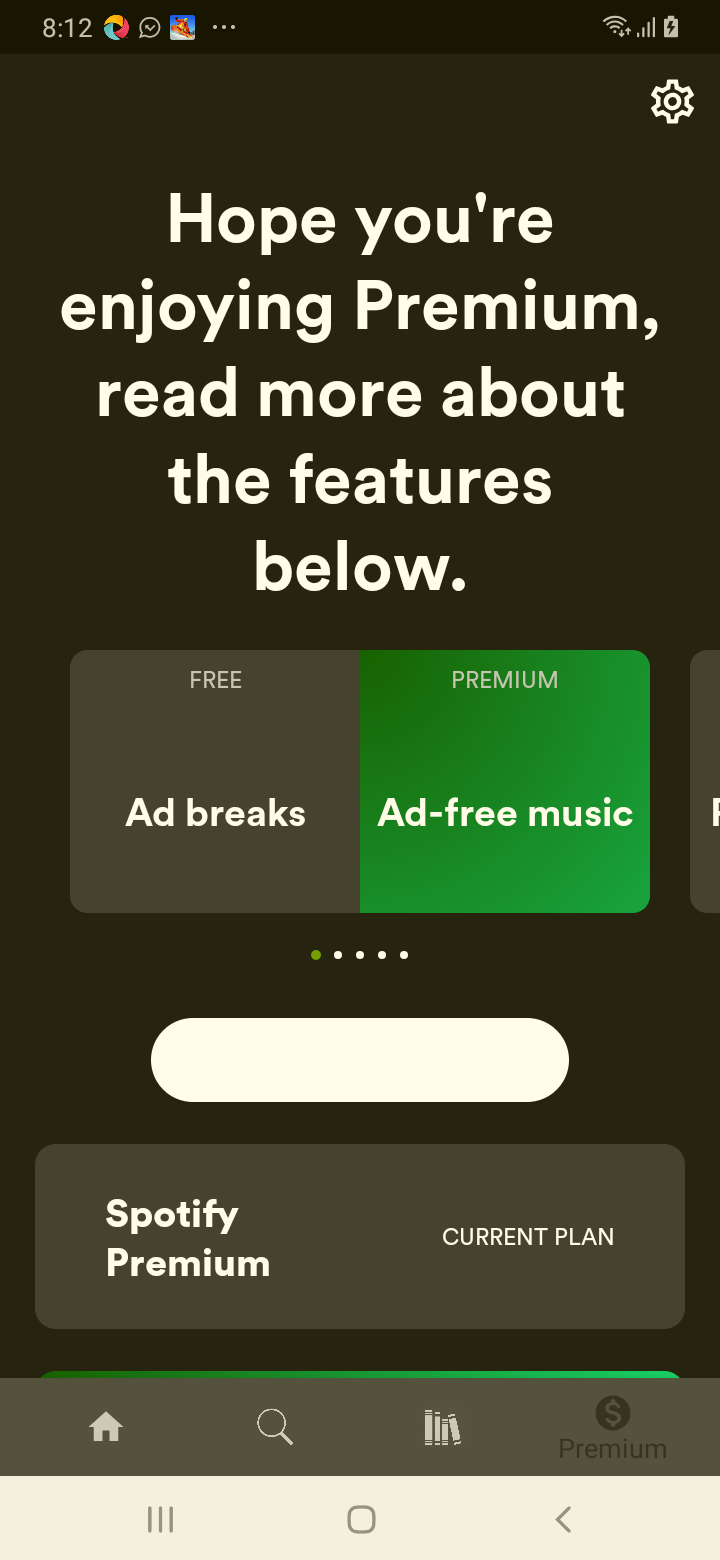
Checking get premium button exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] get premium button exists



## Test Teardown

[Fixture] setup::0

# Checking getting premium - passed

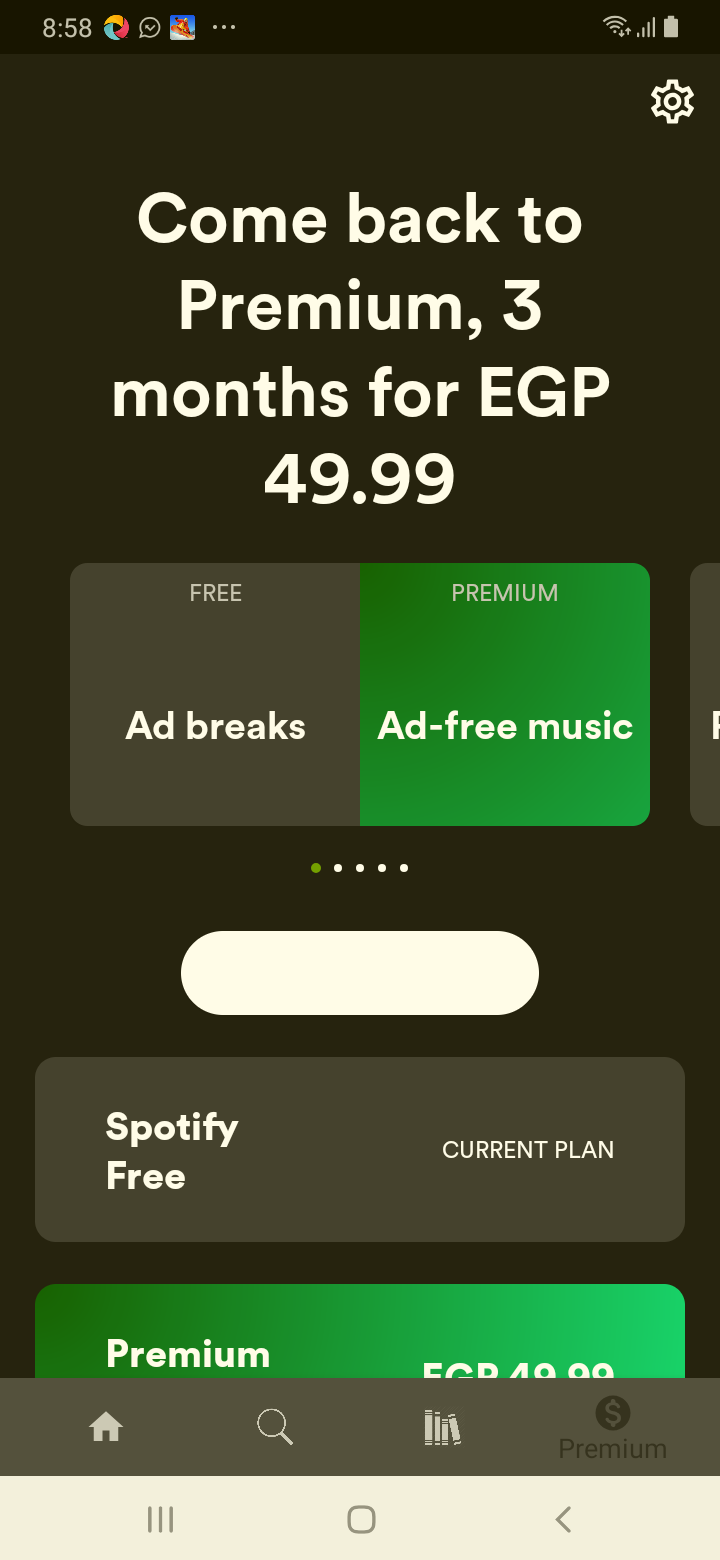
Checking getting premium

## Test Setup

[Fixture] setup

## Test Body

[Attachment] premium activated



## Test Teardown

[Fixture] setup::0

# Checking high audio quality box - passed

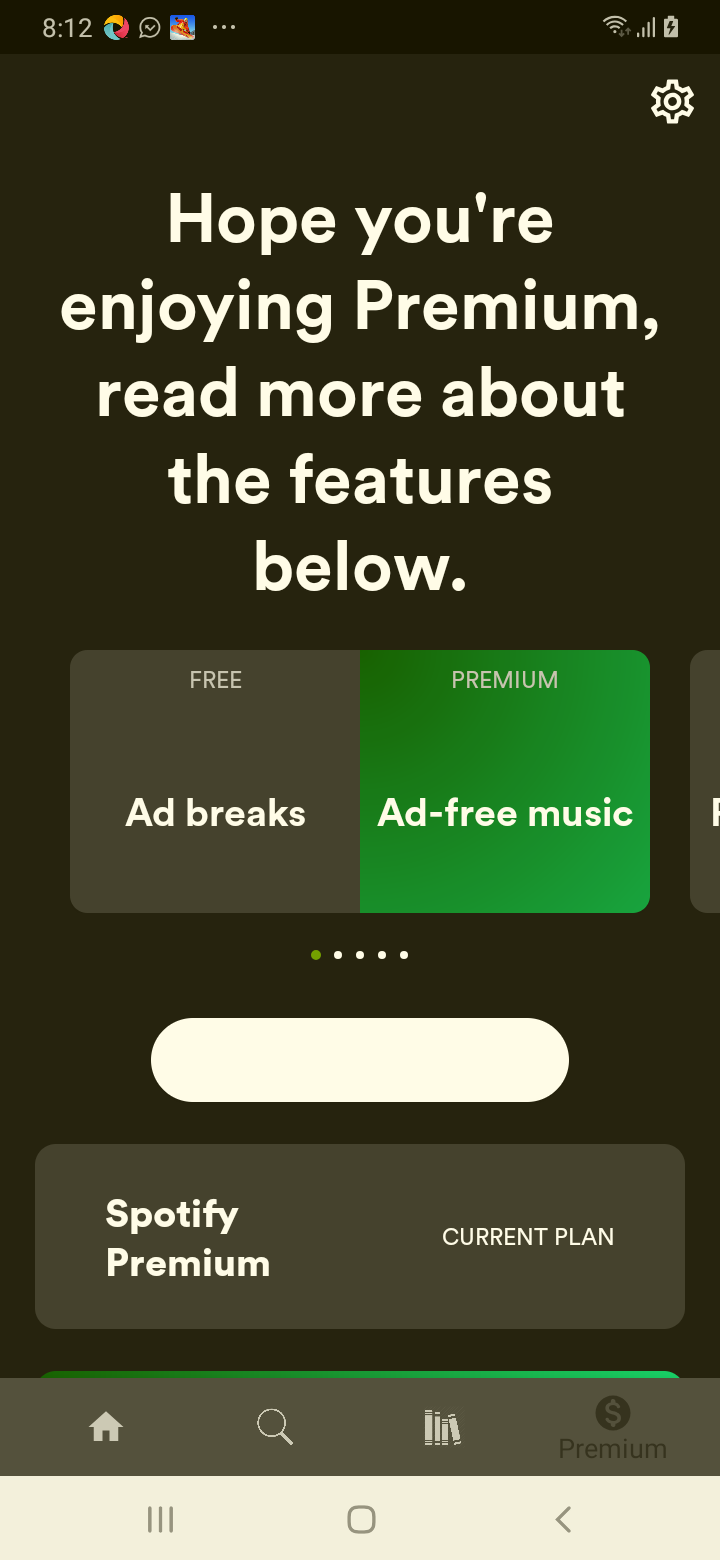
Checking high audio quality exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] high audio quality box exists



## Test Teardown

[Fixture] setup::0

# Checking like - passed

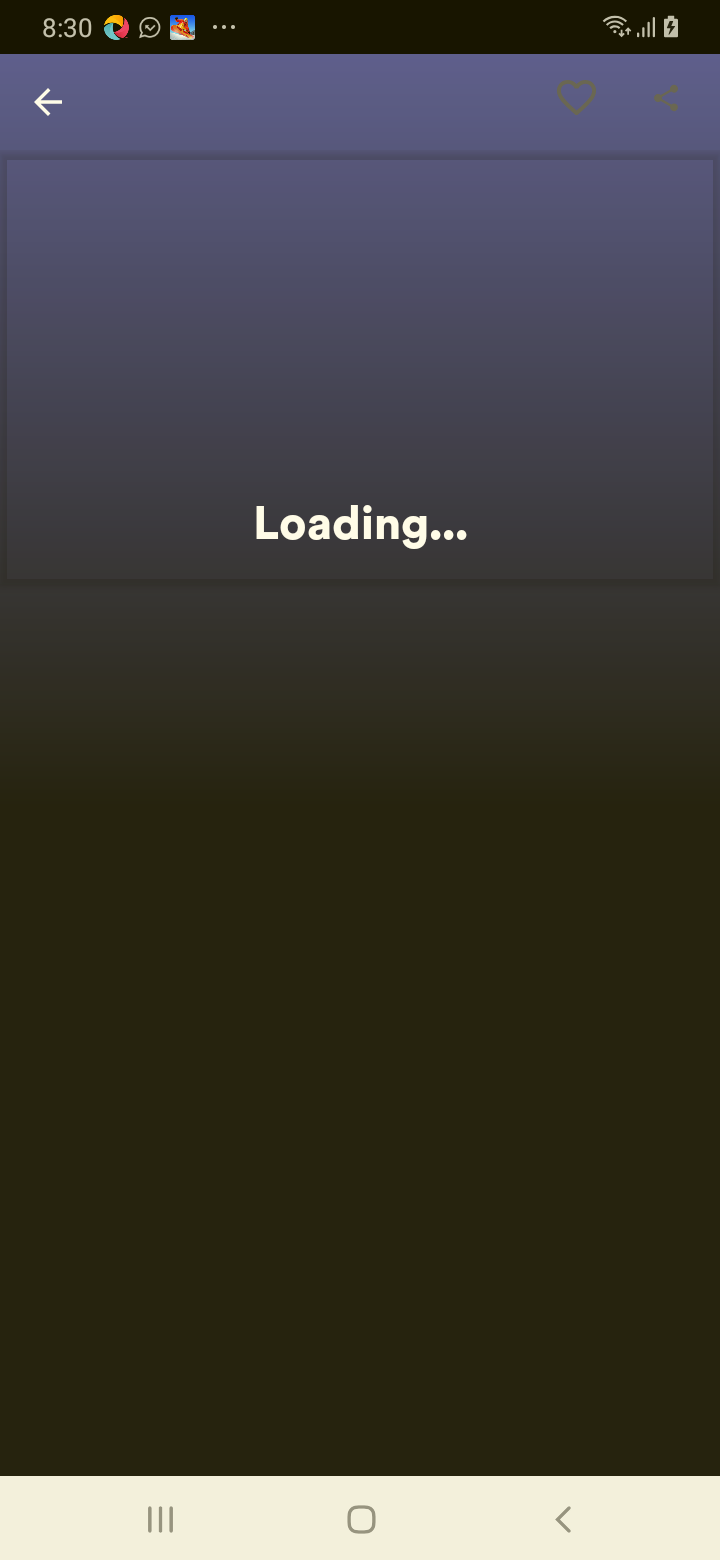
Checking like exists and is click-able

## Test Setup

[Fixture] setup

## Test Body

[Attachment] like exists



## Test Teardown

[Fixture] setup::0

# Checking offline listening box - passed

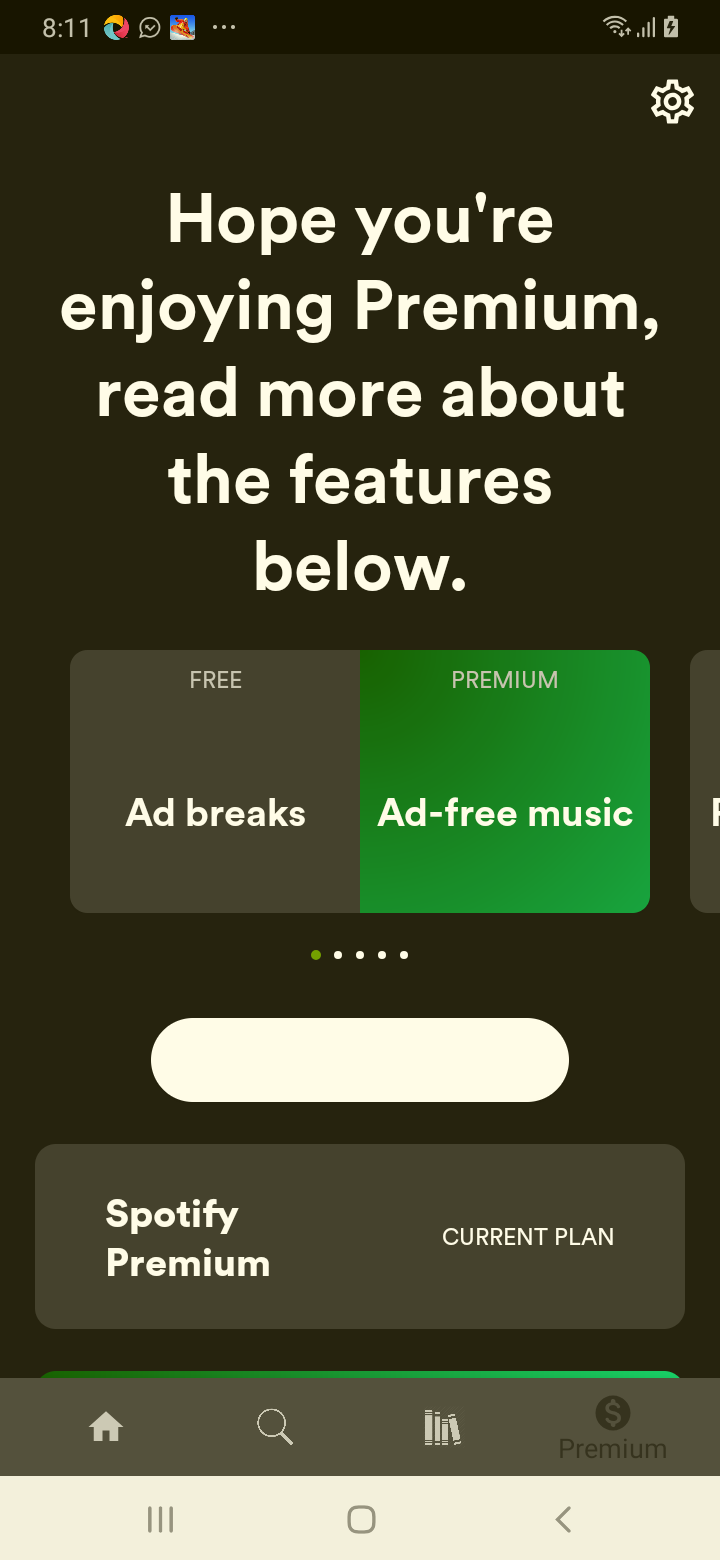
Checking offline listening exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] offline listening box exists



## Test Teardown

[Fixture] setup::0

# Checking play in shuffle box - passed

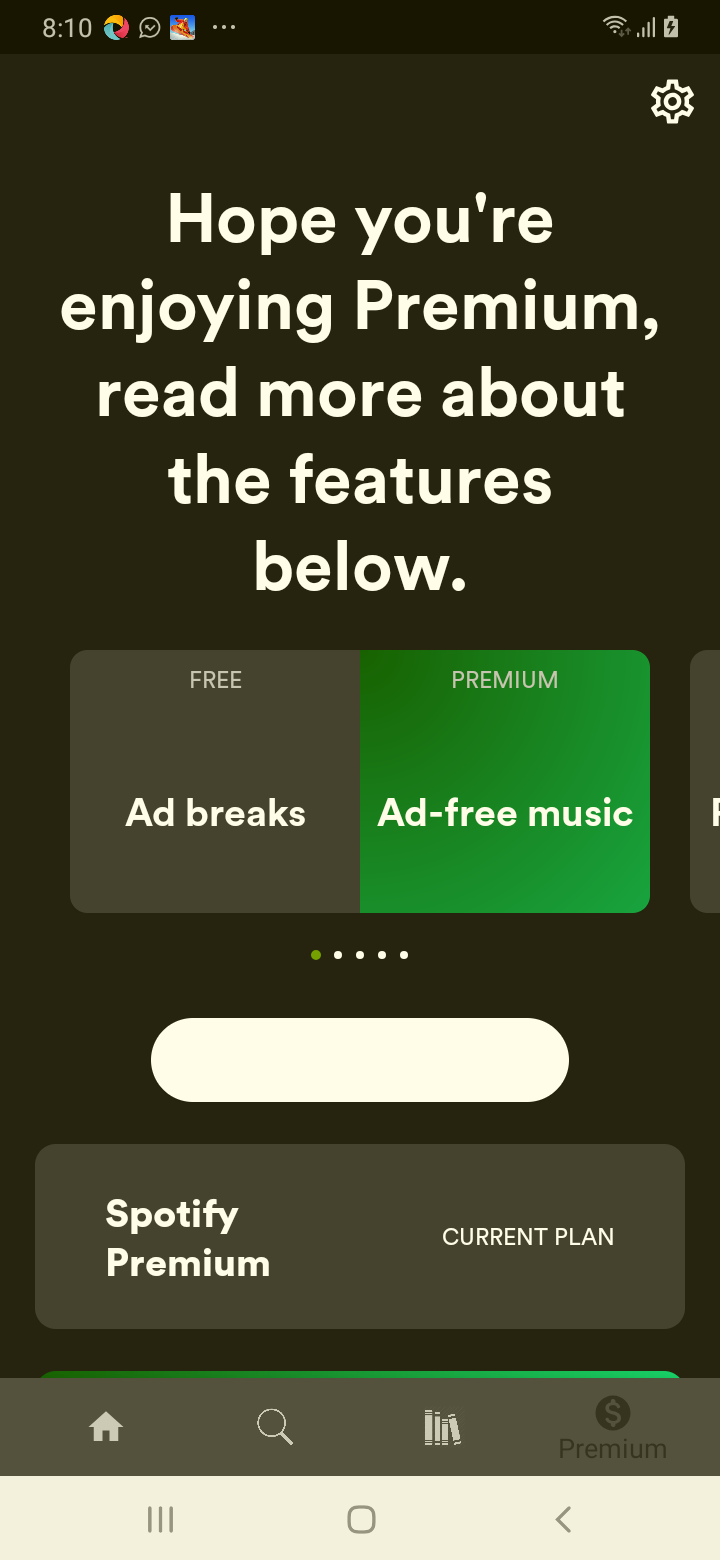
Checking play in shuffle box exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] play in shuffle box exists



## Test Teardown

[Fixture] setup::0

# Checking premium individual box - passed

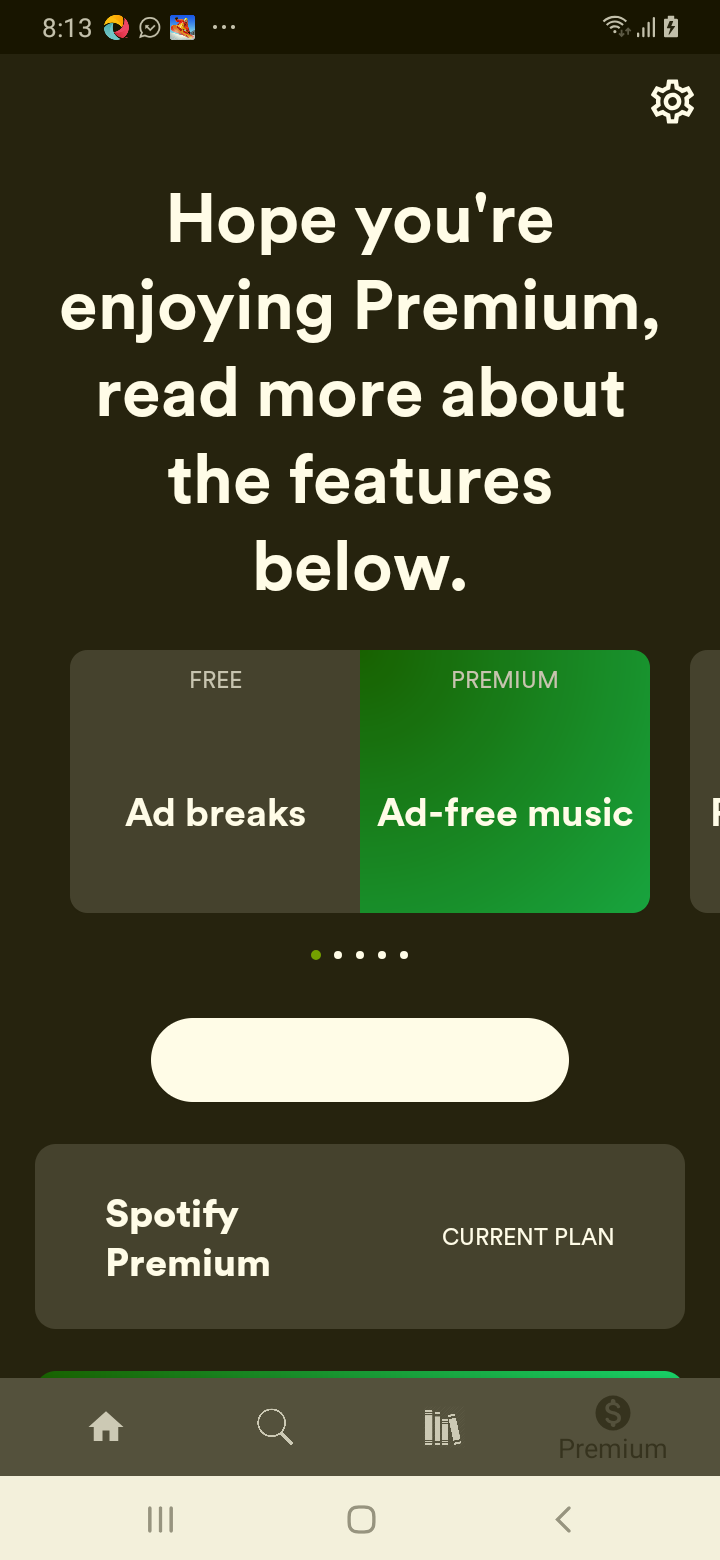
Checking premium individual box exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] premium individual box exists



## Test Teardown

[Fixture] setup::0

# Checking save button - passed

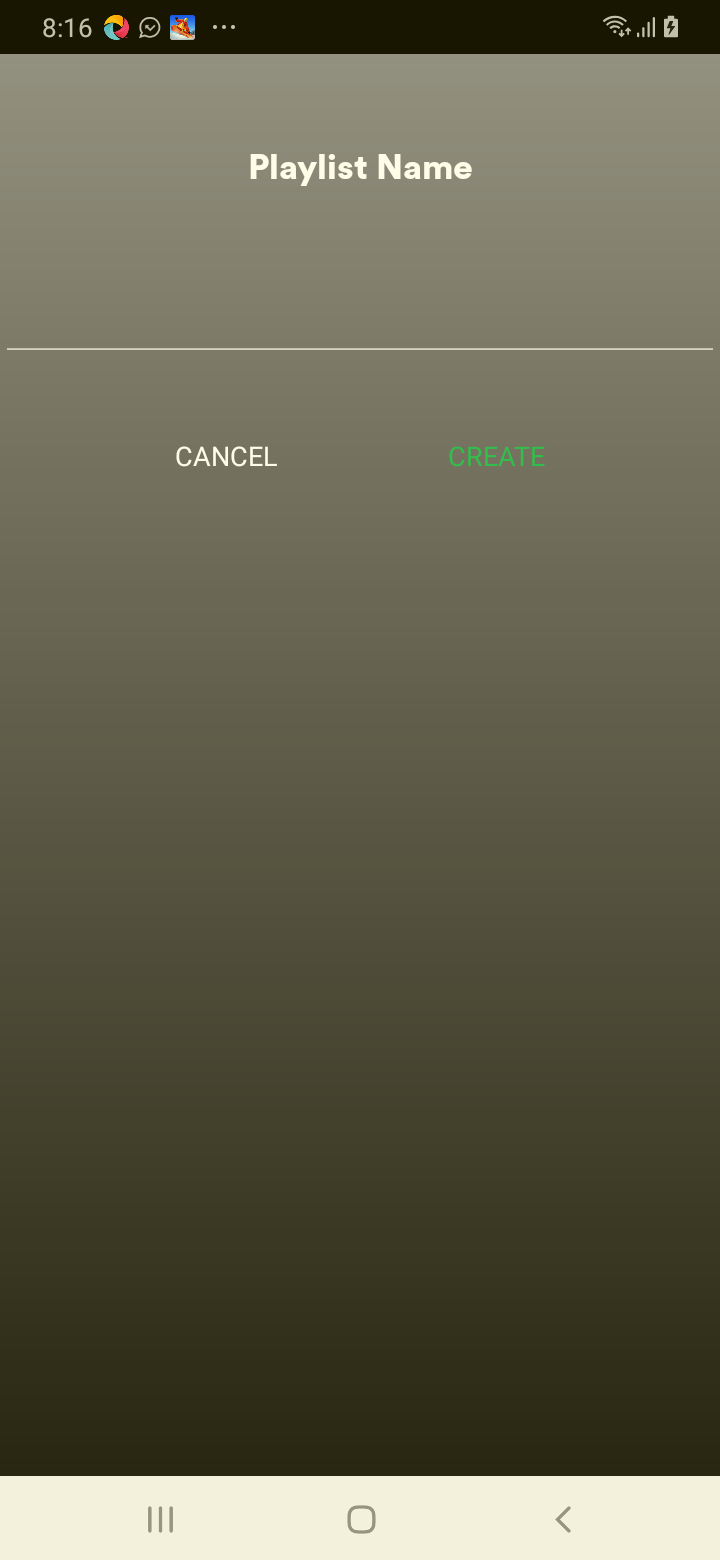
Checking save button exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] save button exist



## Test Teardown

[Fixture] setup::0

# Checking share - passed

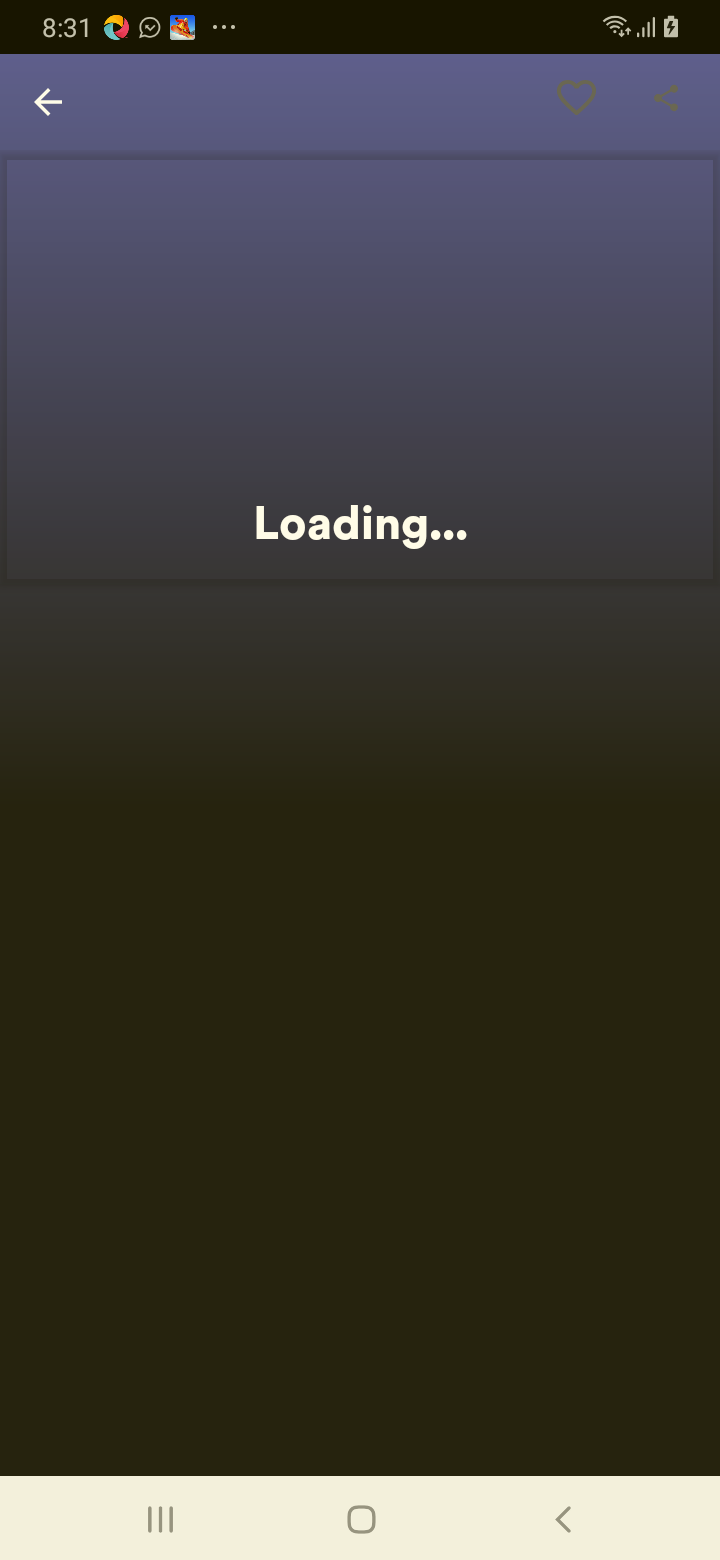
Checking share exists and is click-able

## Test Setup

[Fixture] setup

## Test Body

[Attachment] share exists



## Test Teardown

[Fixture] setup::0

# Checking the add song - passed

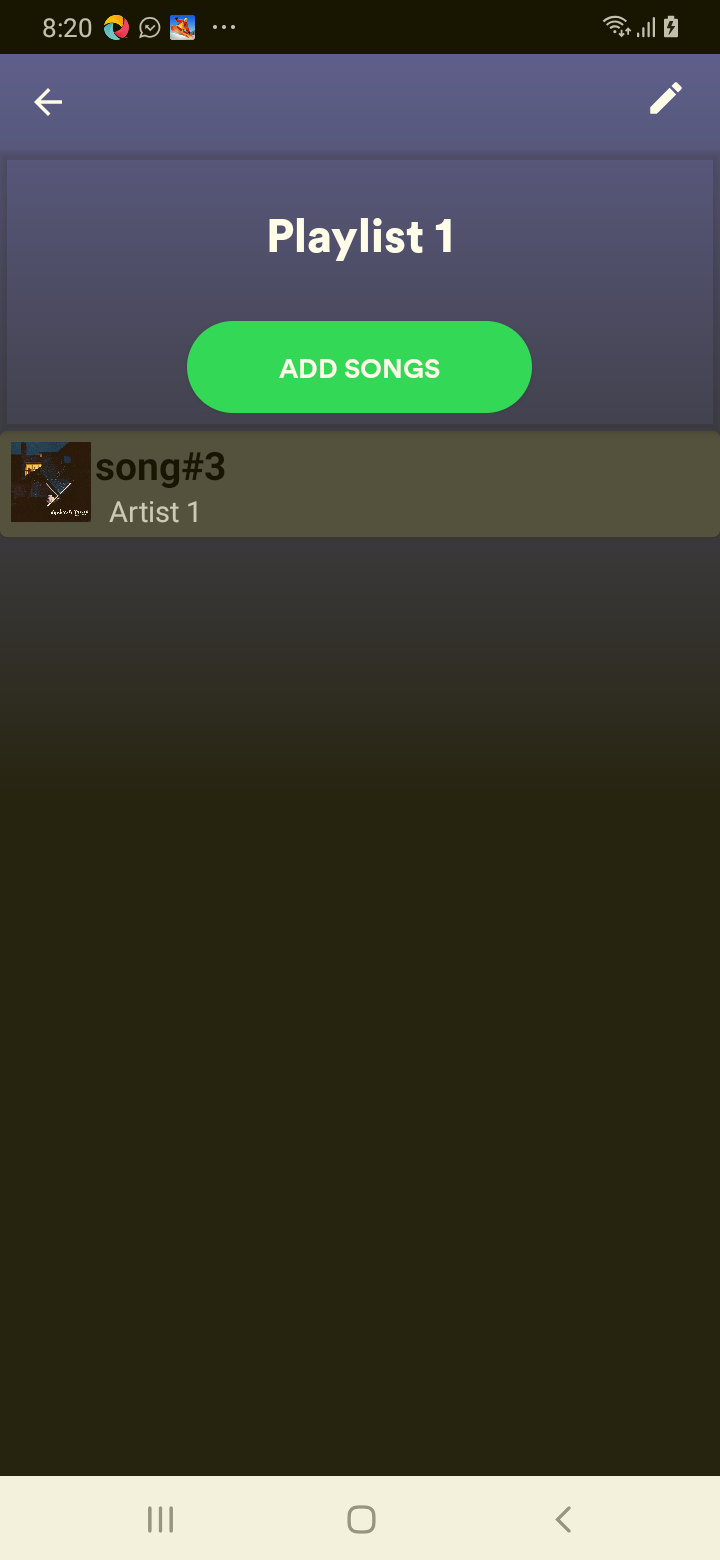
Checking the add song exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] add song exists



## Test Teardown

[Fixture] setup::0

# Checking the edit - passed

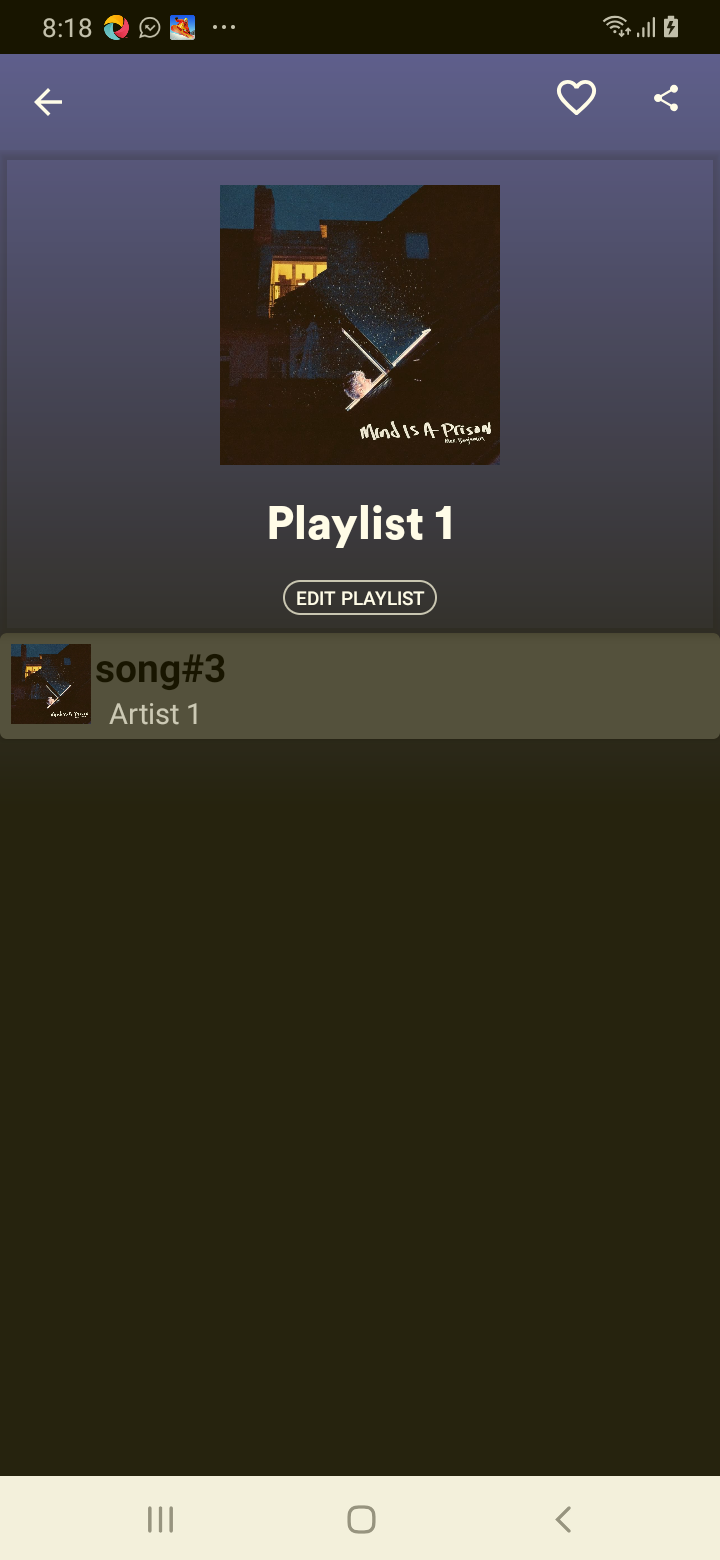
Checking the edit exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] edit exists



## Test Teardown

[Fixture] setup::0

# Checking the edit playlist - passed

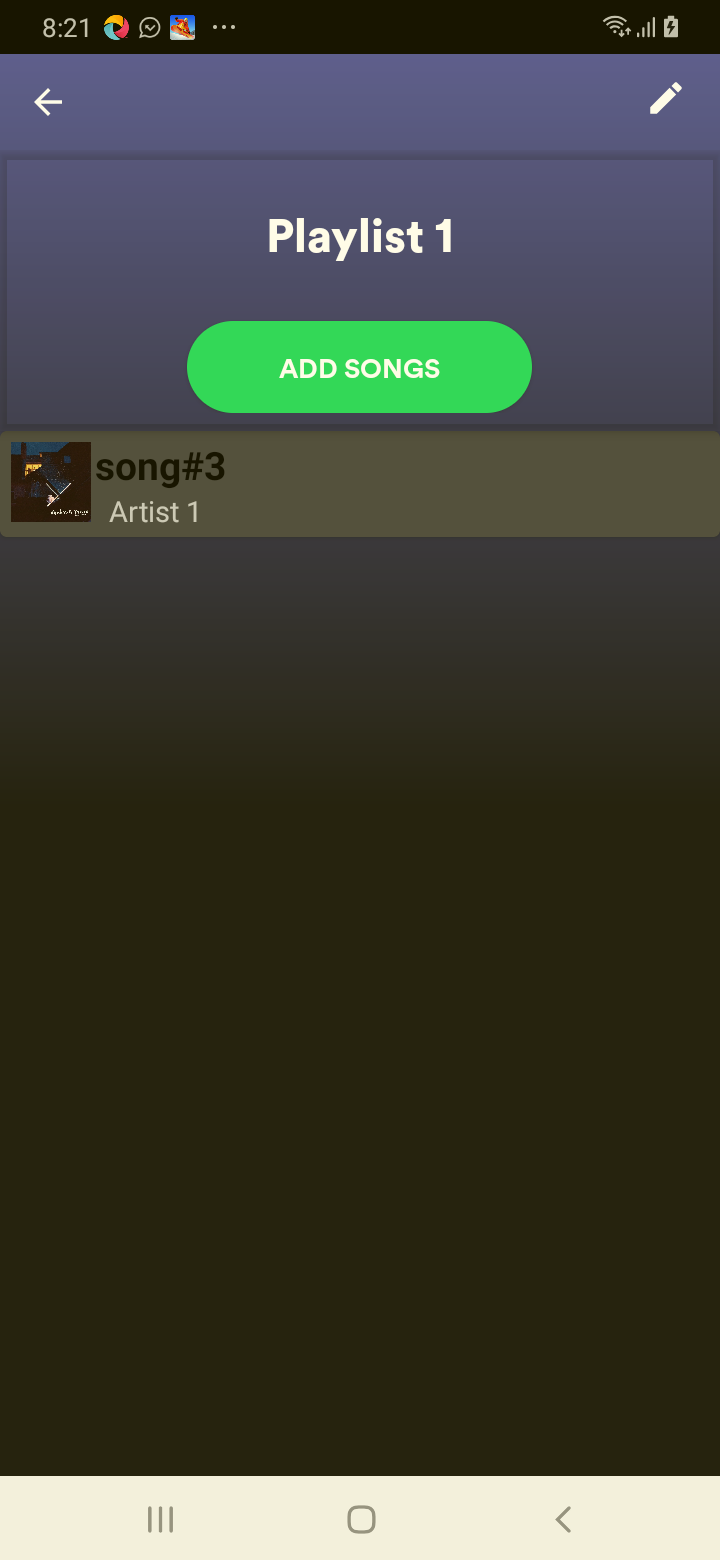
Checking the edit playlist exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] edit playlist exists



## Test Teardown

[Fixture] setup::0

# Checking the first playlist - passed

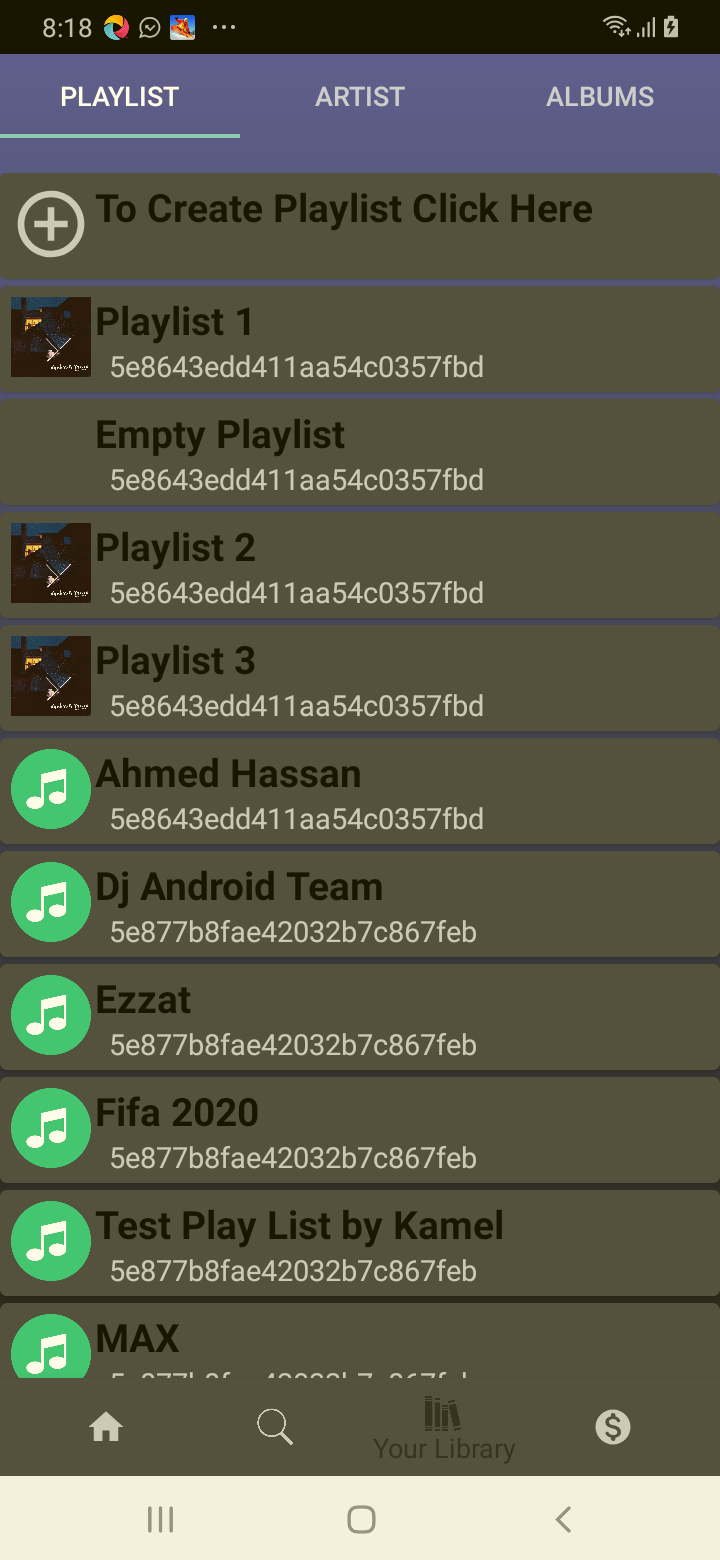
Checking the first playlist exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] first play list exists



## Test Teardown

[Fixture] setup::0

# Checking the like - passed

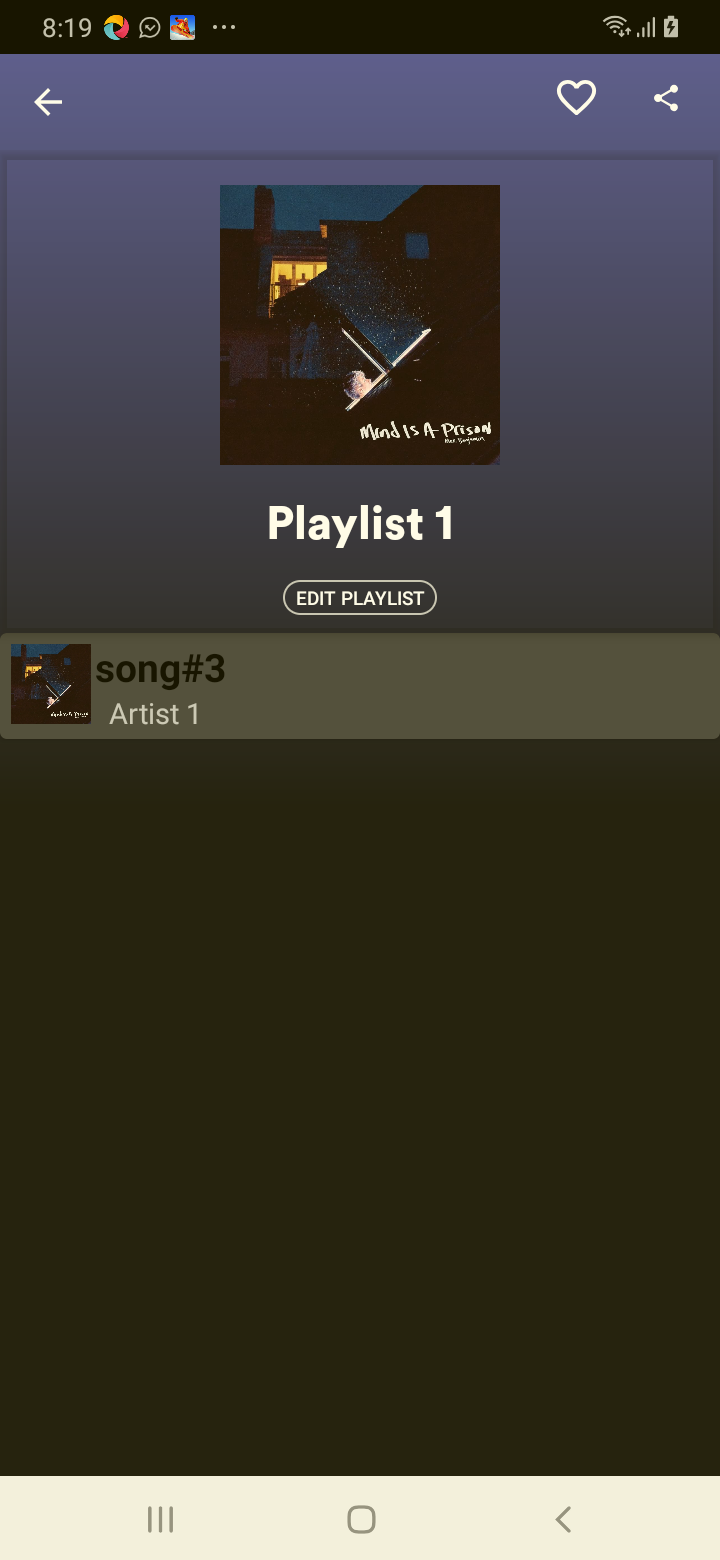
Checking the like exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] like exists



## Test Teardown

[Fixture] setup::0

# Checking the share - passed

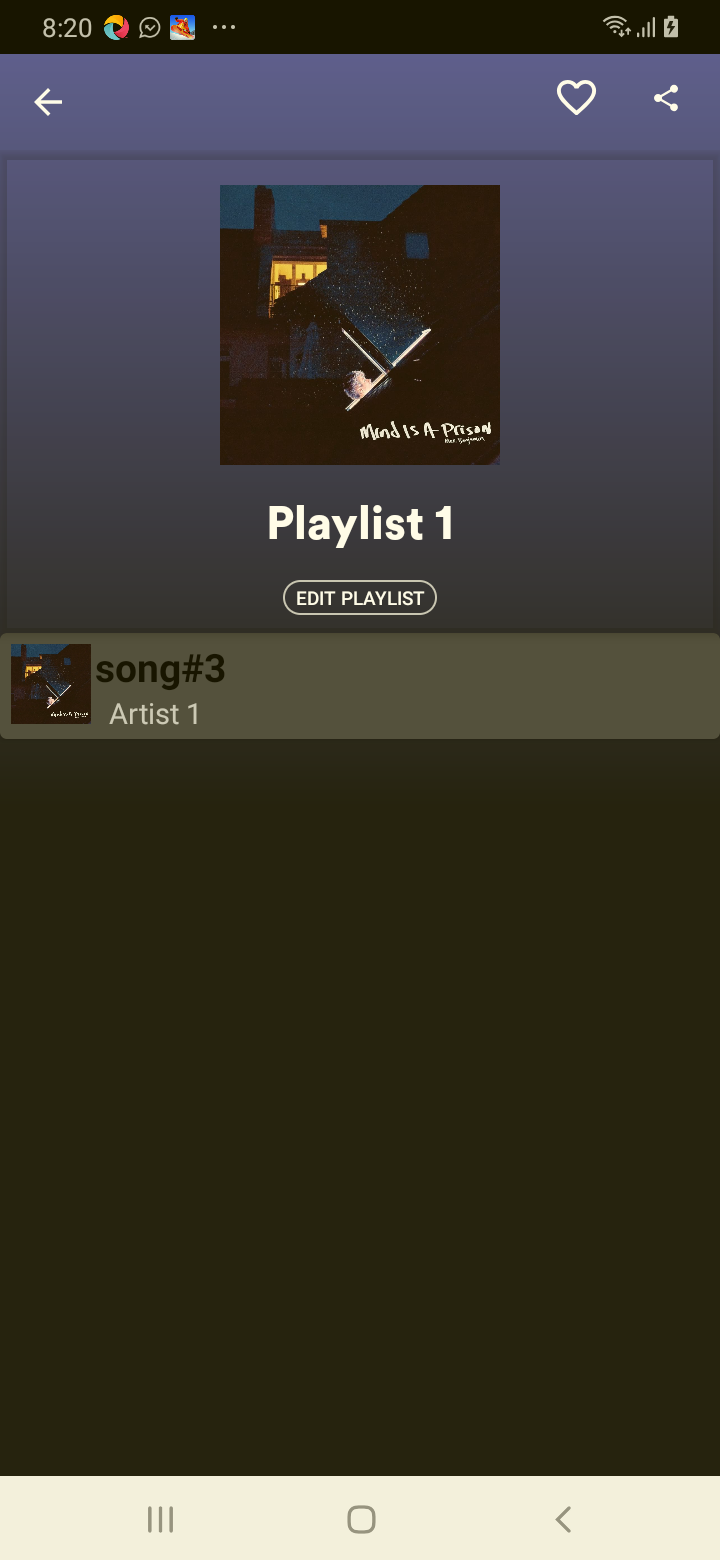
Checking the share exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] share exists



## Test Teardown

[Fixture] setup::0

# Checking unlimited skips box - passed

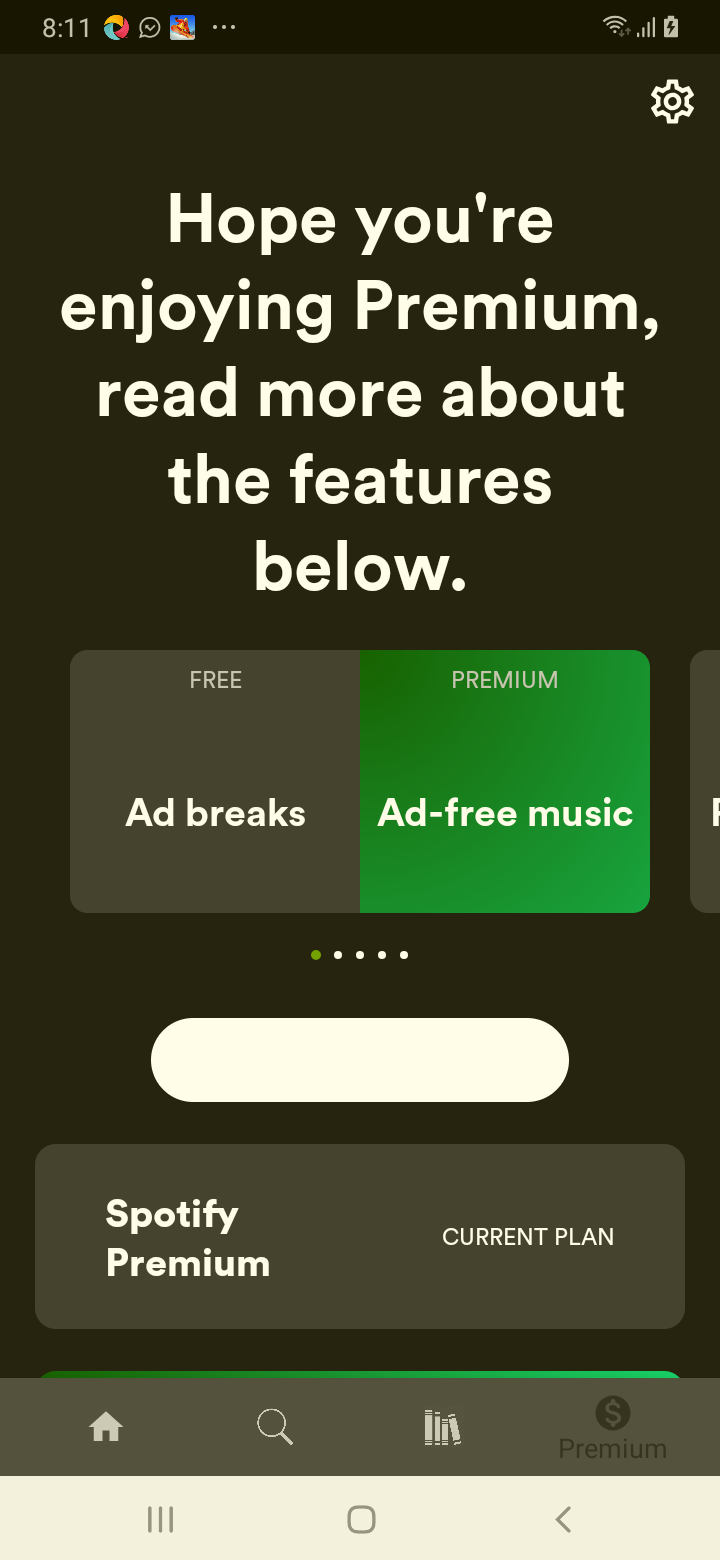
Checking unlimited skips exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] unlimited skips box exists



## Test Teardown

[Fixture] setup::0

# changes the user email to correct email - passed

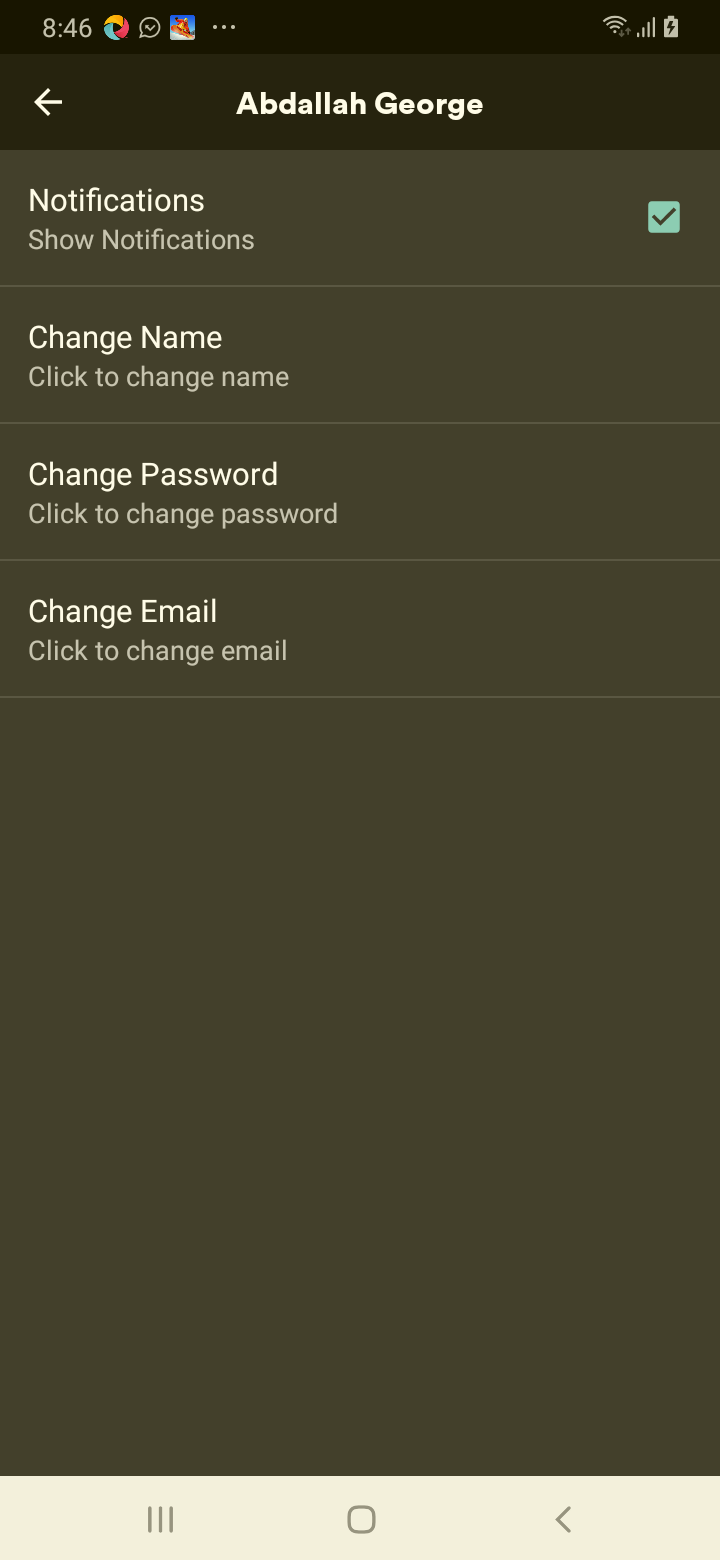
changes the user email to correct email

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# changes the user name to correct name - passed

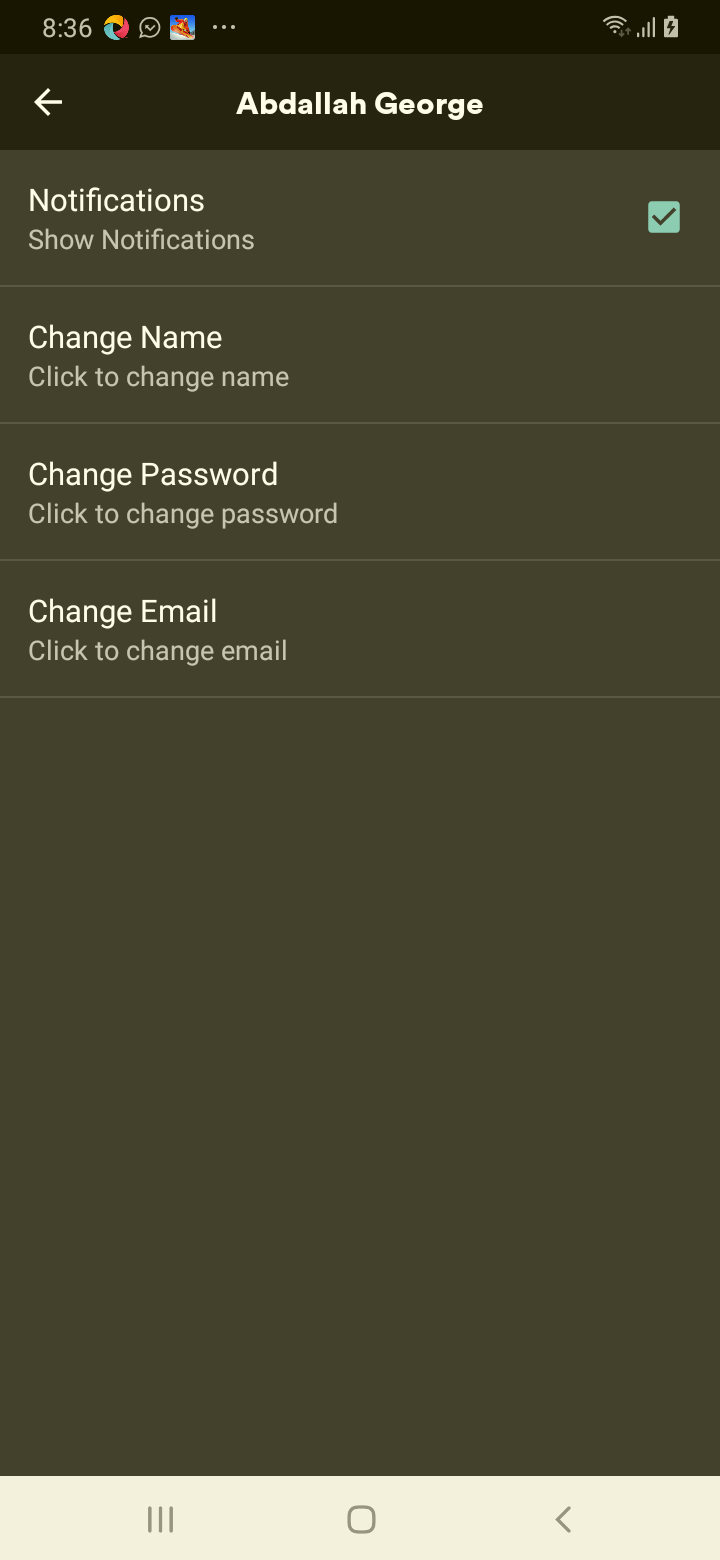
changes the user name to correct name

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# changes the user password to correct password - passed

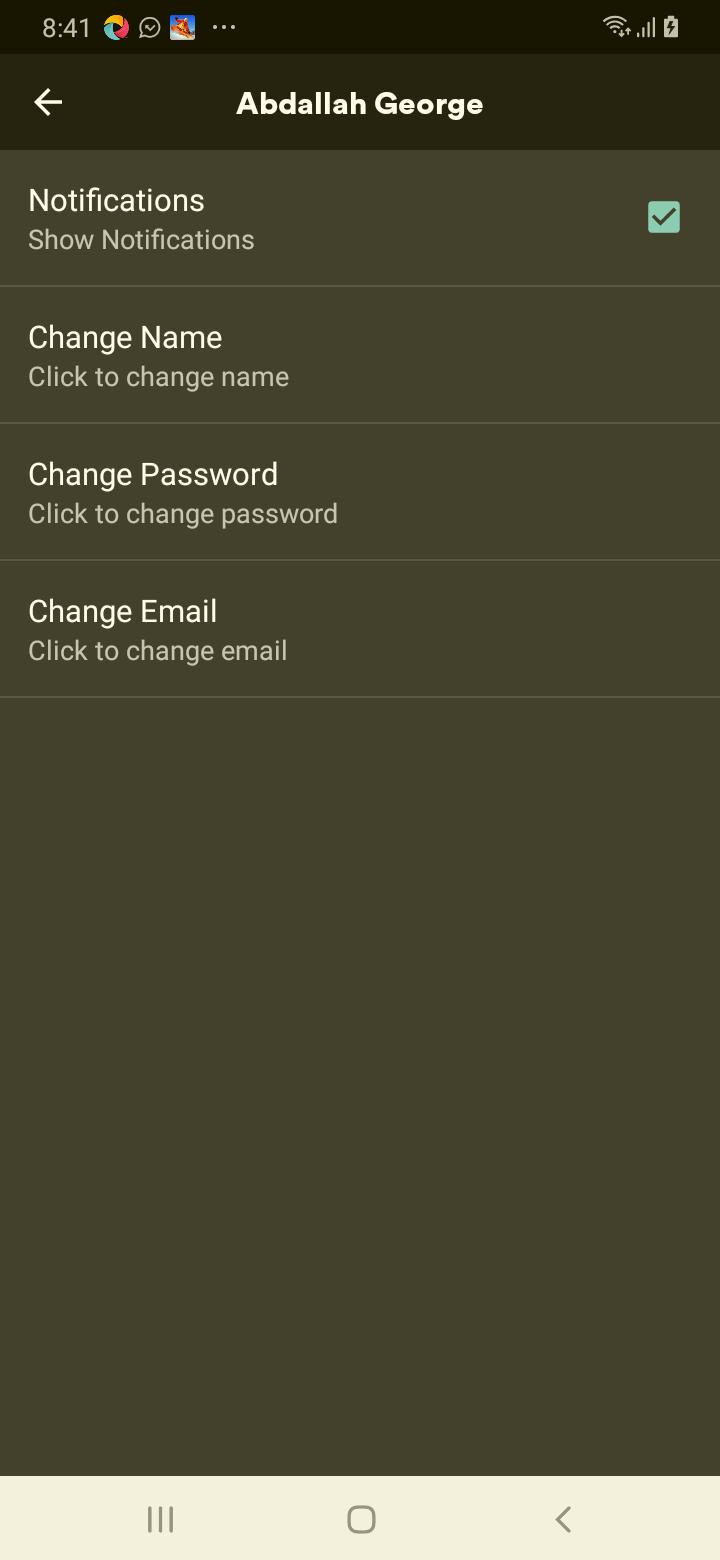
changes the user password to correct password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# checks change email button - passed

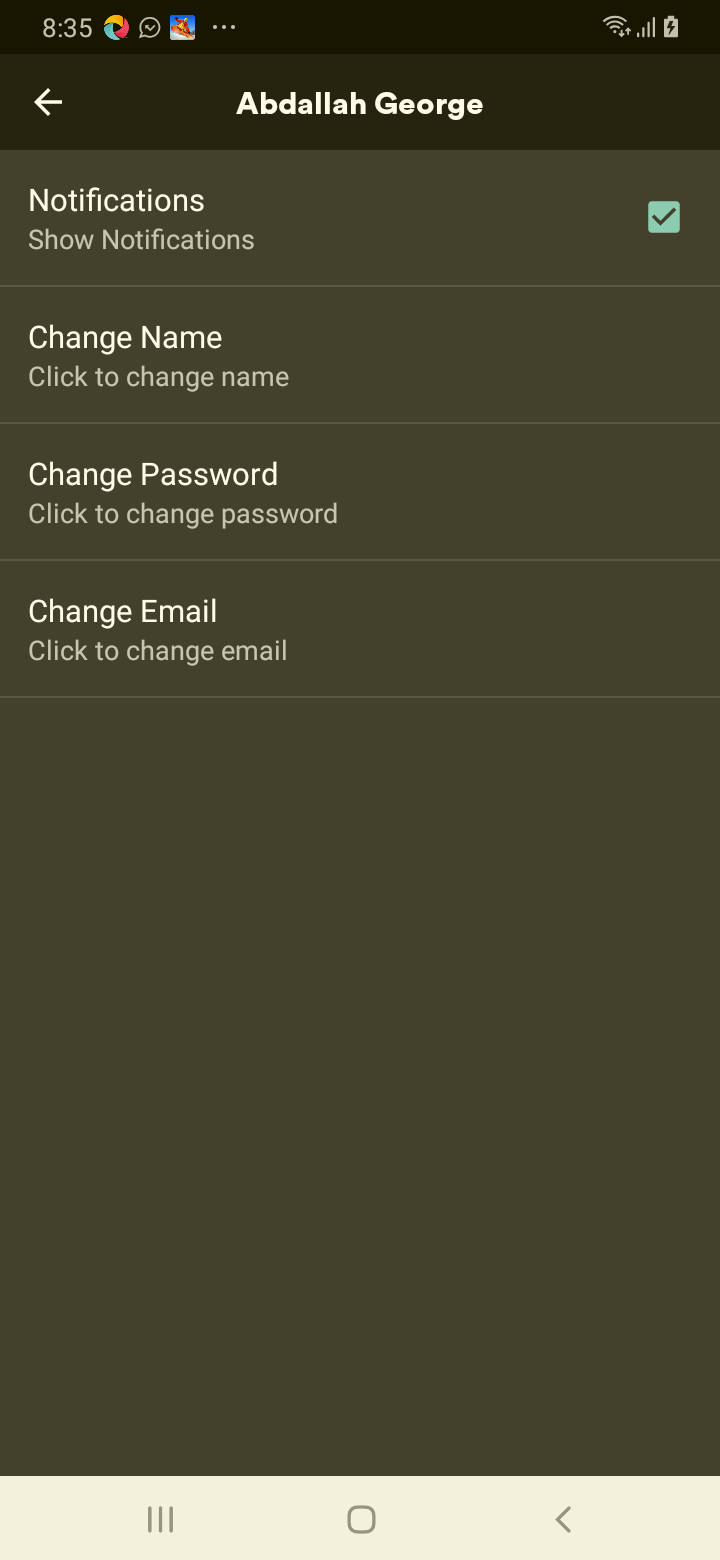
checks change email button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# checks change password button - passed

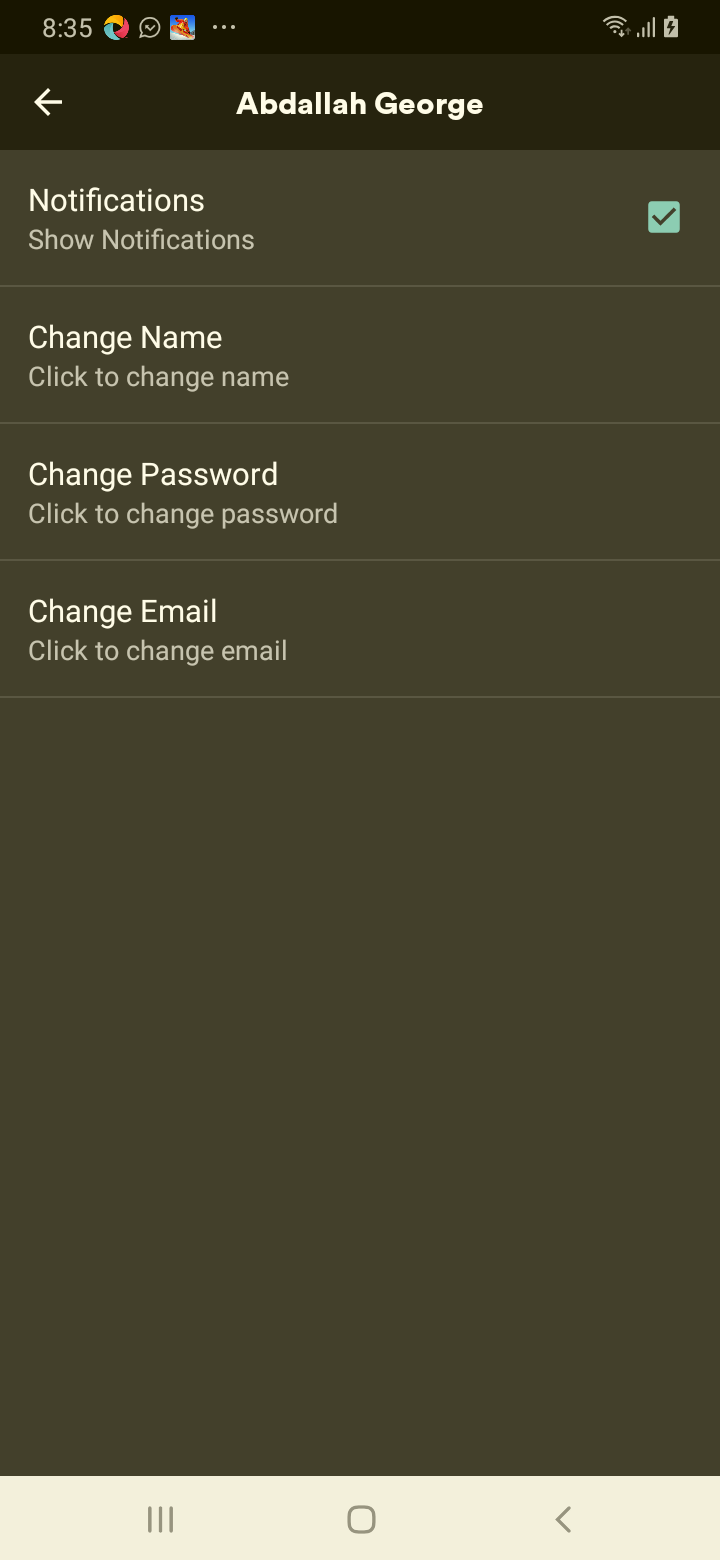
checks change password button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# checks change user name button - passed

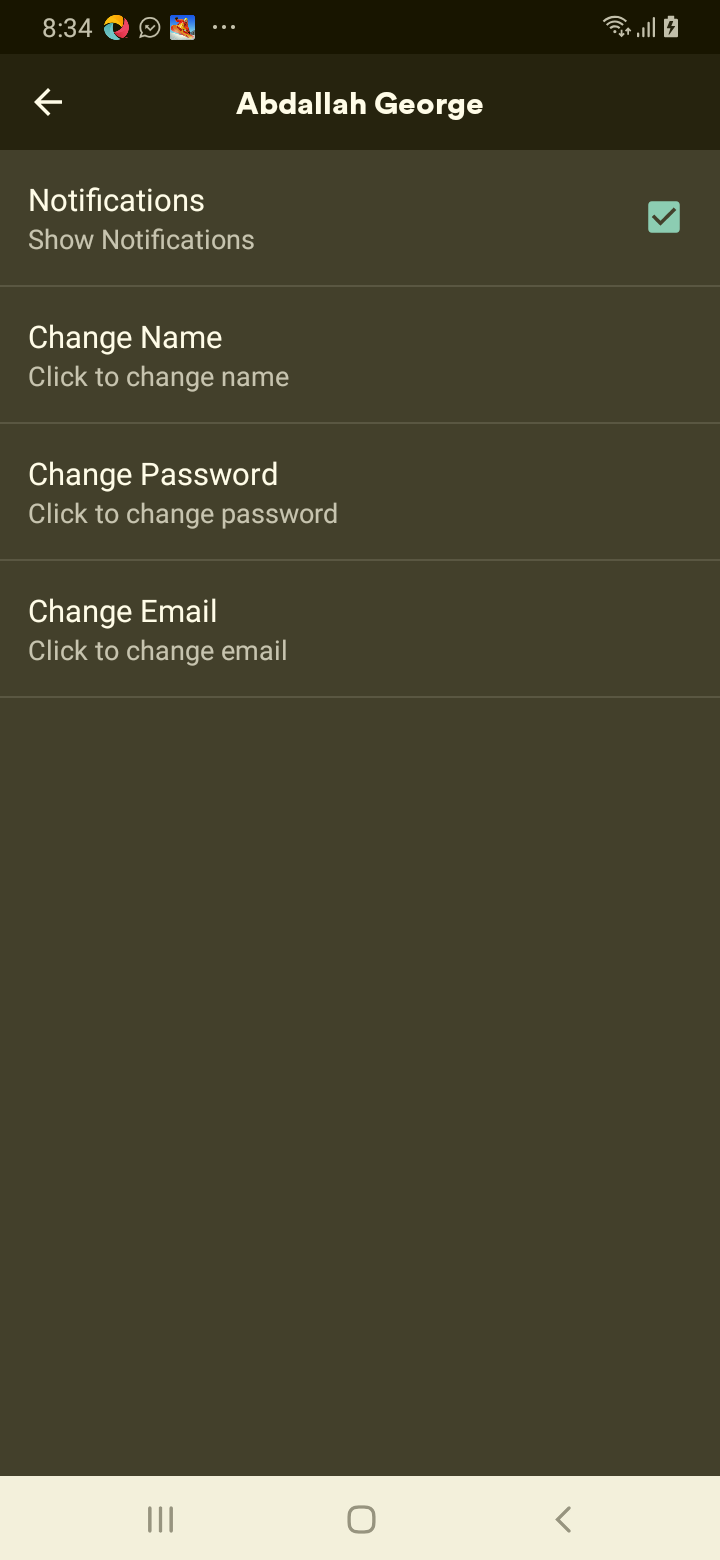
checks change user name button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# checks the edit profile exists - passed

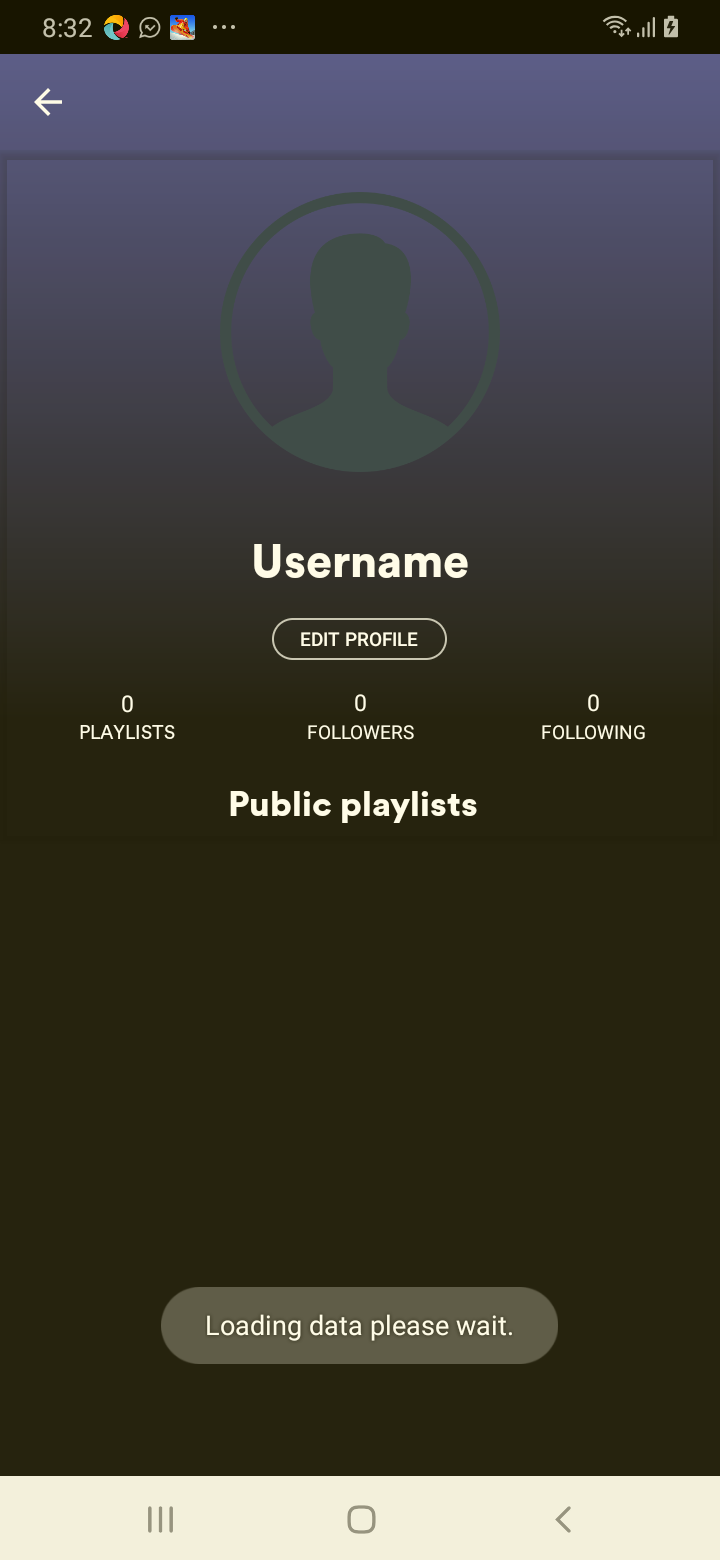
checks the edit profile exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# checks the first playlist exists - passed

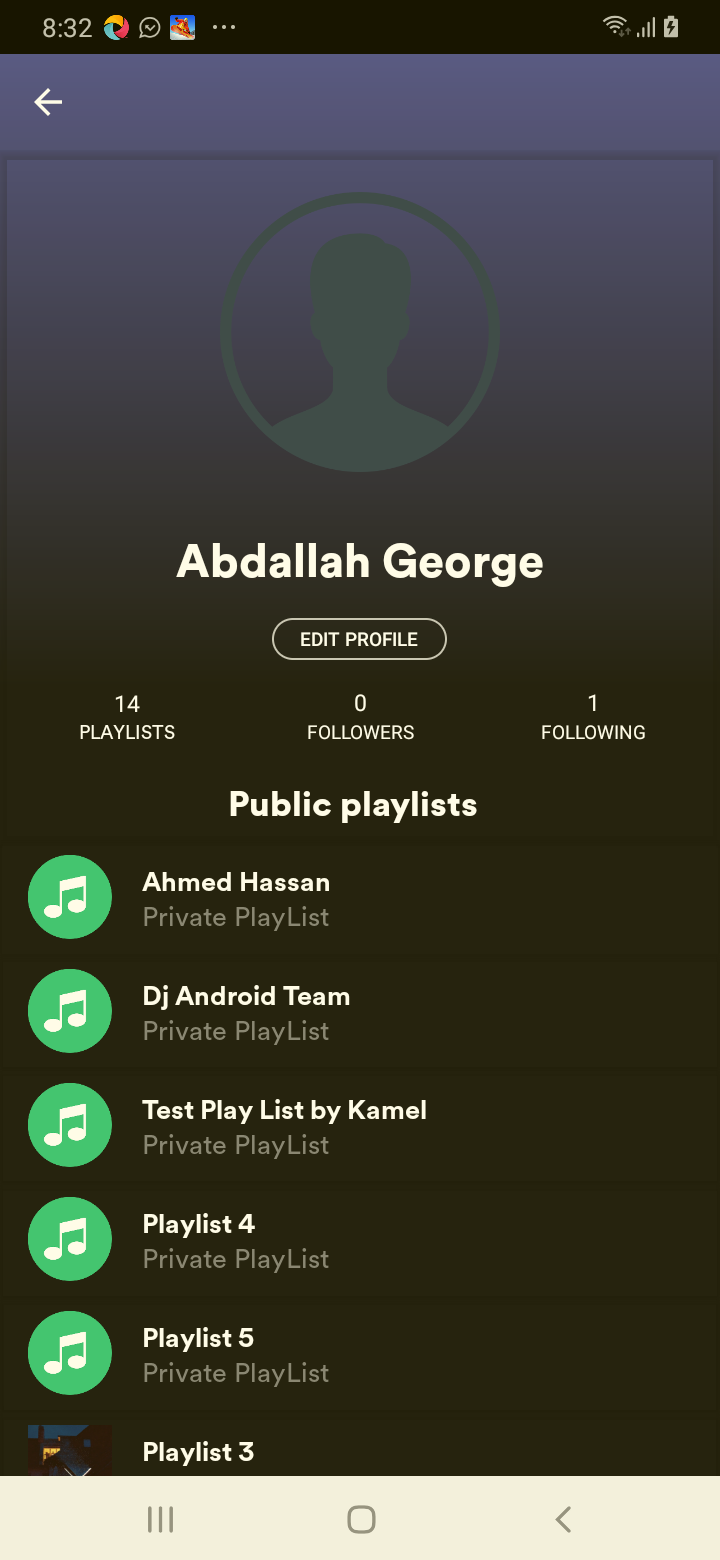
checks the first playlist exists

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0

# checks the notifications checkbox is clickable - passed

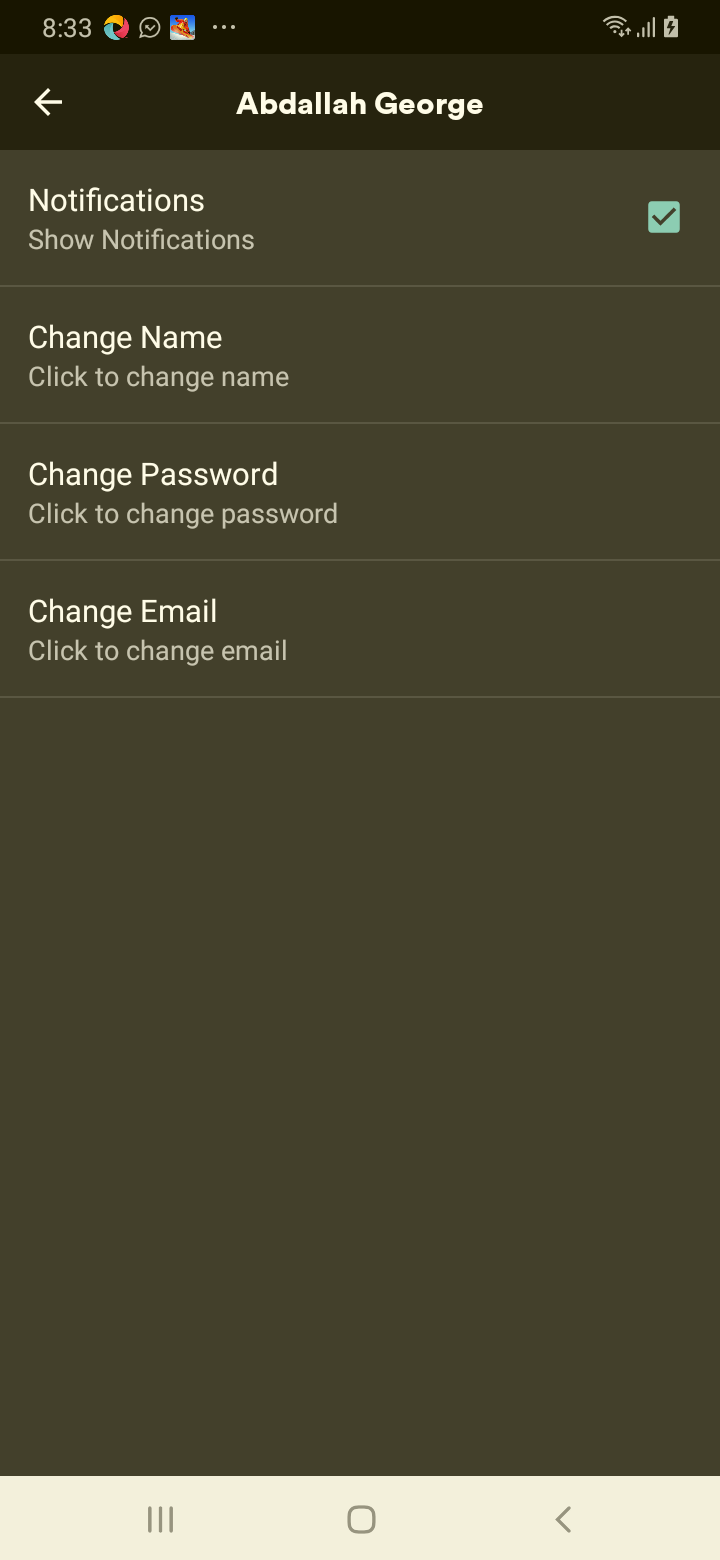
checks the notifications checkbox is clickable

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Passed



## Test Teardown

[Fixture] setup::0