Requirements

Project for Team Project in Java classes

Team: AXOLOTL

Functional REQUIREMENTS

1. [User sign up, successful 2](#_Toc56792777)
2. [The user logs in, successful 3](#_Toc56792778)
3. [The user logs in, with password reminder 4](#_Toc56792779)
4. [The user logs in, unsuccessful 3 times 5](#_Toc56792780)
5. [The user logs out, successful 6](#_Toc56792781)
6. [The user logs out, automate 7](#_Toc56792782)
7. [User creates a post, successful 8](#_Toc56792783)
8. [User edit a post, successful 9](#_Toc56792784)
9. [User deletes a post, successful 10](#_Toc56792785)
10. [User edit his username, successful 11](#_Toc56792786)
11. [User edit date of birth, successful 12](#_Toc56792787)
12. [User edit email, successful 13](#_Toc56792788)
13. [User edit password, successful 14](#_Toc56792789)
14. [Creating a post open to everyone 15](#_Toc56792790)
15. [User browses posts of users he follows 16](#_Toc56792791)
16. [User browses posts of ALL userS 17](#_Toc56792792)
17. [Unfollow option from home page 18](#_Toc56792793)
18. [follow option from discover page 19](#_Toc56792794)
19. [Unfollow option from discover page 20](#_Toc56792795)

|  |  |
| --- | --- |
| **Title:** | User sign up, successful |
| ID: | FR-001 |
| **Description:**  The user wants to sign up. He has no account, goes to the start page, where he can see a welcome form with buttons “Sigh up” and “Log in”. The user clicks “Sign up” and sees a registration form with fields. After completing the form, the user clicks “Sign up” and gets the result: a page with a timeline and navigation buttons. | |
| **The initial conditions:**  The user is not registered.  The user is not authorized. | |
| **Acceptance criteria:**  User has an account, is authorized and can make a post. | |
| **Main scenario:**  1. The user wants to sign up and goes to the start page, he chooses “Sign up”.  2. The system requests the username, the email, the password, and confirm password.  3. The user enters all data and clicks “Sign up”.  4. The system verifies data and creates an account for the user.  5. The system starts a login session and displays a welcome page, where the user can make a post. | |
| **Alternative scenarios and extensions:**  2. The system requests the username, the email, the password, and confirm password:  a. If the username is not unique, the system sends the message “Choose another name”.  b. Redirection to the field “Name”.  c. The user completes the field again.  d. If the email address is already in the database, the system sends the message “This email is already in use”.  e. Redirection to the field “Email”.  f. The user completes the field again.  g. If the password is not strong enough, the system sends the message with password requirements.  h. Redirection to the field “Password”.  i. The user completes the field again.  j. If the confirmed password is not the same as the created password, the system sends the message “Passwords are different”.  k. Redirection to the field “Confirm password”.  l. The user completes the field again. | |

|  |  |
| --- | --- |
| **Title:** | The user logs in, successful |
| ID: | FR-002 |
| **Description:**  The user wants to log in to the MicroBlog. The user already has an account and all login information, he goes to the start page with buttons “Sigh up” and “Log in”. Then the user clicks “Log in” and sees a login form with fields “Email” and “Password”. After he is completing the form and clicks “Log in”, he gets the result: a page with a timeline and navigation buttons. | |
| **The initial conditions:**  The user is registered.  The user is not authorized. | |
| **Acceptance criteria:**  The user is authorized and can make a post. | |
| **Main scenario:**  1. The user wants to log in and goes to the start page, he chooses “Log in”.  2. The system requests the username and password.  3. The user enters his username and password and clicks “Submit”.  4. The system verifies the username and password.  5. The system starts a login session and displays a welcome page, where the user can make a post. | |
| **Alternative scenarios and extensions:**  4. The system verifies the username and password:  a. If the username is invalid, the system sends a message to the user.  b. Redirection to step 2.  c. The user completes the form again and do it right.  4. The system verifies the username and password:  a. If the password is invalid, the system sends a message to user that he has only 3 attempts.  b. Redirection to step 2.  c. The user completes the form again and do it right. | |

|  |  |
| --- | --- |
| **Title:** | The user logs in, with password reminder |
| ID: | FR-003 |
| **Description:**  The user wants to log in to the MicroBlog. The user already has an account but does not remember all login information. He goes to the start page. When the user clicks “Log in” and sees a login form with fields “Email” and “Password” and a special label “I don’t remember the password”. The user clicks a label and sees a form “Please, input your email”. After the user fills a field and clicks “Submit”, he will see a message “Password was sent to this email, check it and try to log in with new password” and the page will be redirected to the start page. The system generates a new password and sends it to the user. The user checks email and tries to log in with a new password. | |
| **The initial conditions:**  The user is registered.  The user is not authorized. | |
| **Acceptance criteria:**  The user is authorized and can make a post. | |
| **Main scenario:**  1. The user wants to log in and goes to the start page, he chooses “Log in”.  2. The system requests the username and password.  3. The user chooses the label “I don’t remember the password”.  4. The system requests the user’s email.  5. The user input email and press “Submit”.  6. The system sends a new password to this email and redirects the user to the start page.  7. The user checks his email and gets a new password.  7. The user goes to the start page, he chooses “Log in”.  9. The user enters his username and password and clicks “Submit”.  10. The system verifies the username and password.  11. The system starts a login session and displays a welcome page, where the user can make a post. | |
| **Alternative scenarios and extensions:** | |

|  |  |
| --- | --- |
| **Title:** | The user logs in, unsuccessful 3 times |
| ID: | FR-004 |
| **Description:**  The user wants to log in to the MicroBlog. The user already has an account but does not remember all login information. He goes to the start page and chooses “Log in”. After three unsuccessful attempts to enter the password, the system sends a message “Please, change your password” and redirects the user to a form “Please, input your email”. After the user fills a field and clicks “Submit”, he will see a message “Password was sent to this email, check it and try to log in with new password” and the page will be redirected to the start page. The system generates a new password and sends it to the user. The user checks the email and tries to log in with a new password. | |
| **The initial conditions:**  The user is registered.  The user is not authorized. | |
| **Acceptance criteria:**  The user is authorized and can make a post. | |
| **Main scenario:**  1. The user wants to log in and goes to the start page, he chooses “Log in”.  2. The system requests the username and password.  3. The user enters his username and password and clicks “Submit”.  4. The system verifies the username and password.  5. The password is invalid, the system sends a message to user that he has only 3 attempts and redirects to step 2.  4. After third unsuccessful attempt the system sends a message “Please, change your password” and requests the user’s email.  5. The user input email and press “Submit”.  6. The system sends a new password to this email and redirects the user to the start page.  7. The user checks his email and gets a new password.  7. The user goes to the start page, he chooses “Log in”.  9. The user enters his username and password and clicks “Submit”.  10. The system verifies the username and password.  11. The system starts a login session and displays a welcome page, where the user can make a post. | |
| **Alternative scenarios and extensions:** | |

|  |  |
| --- | --- |
| **Title:** | The user logs out, successful |
| ID: | FR-005 |
| **Description:**  The user wants to log out from the MicroBlog. At any page user can find button “Log out”. When the user clicks it, he sees form with question “Do you want to log out?” and buttons “Yes” and “No”. When the user chooses “Yes”, he gets to the start page. | |
| **The initial conditions:**  The user is authorized.  The user is on any page. | |
| **Acceptance criteria:**  The user is unauthorized, sees the start page. | |
| **Main scenario:**  1. The user is on the any page.  2. The user clicks “Log out”.  3. The system generates the form “Do you want to log out?”.  3. The user chooses “Yes”.  4. The system ends the session and writes all user’s info to the database.  5. The system redirects the user to the start page, user is unauthorized. | |
| **Alternative scenarios and extensions:**  3. The user choose “No”.  4. User stays on this page. | |

|  |  |
| --- | --- |
| **Title:** | The user logs out, automate |
| ID: | FR-006 |
| **Description:**  The user is authorized and stays on any page of the Microblog longer than 10 minutes. After this time the system ends this session and sends a message to the user “You need to log in again”. When the user clicks “Okay”, the system redirects him to the start page. | |
| **The initial conditions:**  The user is authorized.  The user is on the any page. | |
| **Acceptance criteria:**  The user is unauthorized, sees the start page. | |
| **Main scenario:**  1. The user is on the any page.  2. The user does not do anything on this page during 10 minutes.  3. Server ends session and sends a message to the user “You need to log in again”.  4. The user clicks “Okay”.  5. The system redirects the user to the start page, user is unauthorized. | |
| **Alternative scenarios and extensions:** | |

|  |  |
| --- | --- |
| **Title:** | User creates a post, successful |
| ID: | FR-007 |
| **Description:**  The user wants to create a post. He is on the home page. The user clicks to the field, enters the text and posts it. After that, the user sees this post in the timeline. | |
| **The initial conditions:**  The user is registered.  The user is on home page. | |
| **Acceptance criteria:**  User sees his post in the timeline. | |
| **Main scenario:**  1. The user wants to create a post.  2. The user enters the text in the field.  3. The user clicks “Post”.  4. The system writes this data to the database.  5. The post is shown in the timeline. | |
| **Alternative scenarios and extensions:**  2. The user clicks “Post”:  a. No connection with the server, the system sends a message “No connection”.  b. The post is not created and the user can not see it in the timeline. | |

|  |  |
| --- | --- |
| **Title:** | User edit a post, successful |
| ID: | FR-008 |
| **Description:**  The user wants to edit a post. He is on the home page. The user clicks to the post, edits the text and submits it. After that, the user sees this edited post in the timeline. Post is marked as edited, so other users know it was modified. | |
| **The initial conditions:**  The user is registered.  The user is on home page. | |
| **Acceptance criteria:**  User sees his edited post in the timeline. | |
| **Main scenario:**  1. The user wants to edit a post.  2. The user clicks on the post.  3. The user edits the text in the field.  4. The user clicks “Submit”.  4. The system writes this data to the database.  5. The edited post is shown in the timeline. | |
| **Alternative scenarios and extensions:**  2. The user clicks “Submit”:  a. No connection with the server, the system sends a message “No connection”.  b. The post is not edited and the user sees an old post it in the timeline. | |

|  |  |
| --- | --- |
| **Title:** | User deletes a post, successful |
| ID: | FR-009 |
| **Description:**  The user wants to delete a post. He is on the home page. The user clicks on the post, choose “Delete”. After that, the user does not see this post in the timeline. | |
| **The initial conditions:**  The user is registered.  The user is on home page. | |
| **Acceptance criteria:**  User does not see this post in the timeline. | |
| **Main scenario:**  1. The user wants to delete a post.  2. The user chooses the post.  3. The user clicks “Delete”.  4. The system writes this data to the database.  5. The post is not shown in the timeline. | |
| **Alternative scenarios and extensions:**  2. The user clicks “Delete”:  a. No connection with the server, the system sends a message “No connection”.  b. Post is not deleted and the user can see it in the timeline. | |

|  |  |
| --- | --- |
| **Title:** | User edit his username, successful |
| ID: | FR-010 |
| **Description:**  The user wants to edit his information. He has an account, he goes to the profile page, where he sees all the information about the user. The user clicks “Edit” and sees an edit form with fields. After filling out the form, the user clicks “Submit” and sees the edited information about the user. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User sees his edited information. | |
| **Main scenario:**  1. The user wants to change his information and goes to the profile page, he chooses “Edit”.  2. The system shows the username, the email, the date of birth.  3. The user enters another username and clicks “Submit”.  4. The system verifies data and changes the name for the user.  5. The user can see a profile with another name. | |
| **Alternative scenarios and extensions:**  4. The system verifies data and changes the name for the user:  a. If the username is not unique, the system sends the message “Choose another name”.  b. Redirection to the field “Name”.  c. The user completes the field again. | |

|  |  |
| --- | --- |
| **Title:** | User edit date of birth, successful |
| ID: | FR-011 |
| **Description:**  The user wants to edit his information. He has an account, he goes to the profile page, where he sees all the information about the user. The user clicks “Edit” and sees an edit form with fields. After filling out the form, the user clicks “Submit” and sees the edited information about the user. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User sees his edited information. | |
| **Main scenario:**  1. The user wants to change his information and goes to the profile page, he chooses “Edit”.  2. The system shows the username, the email, the date of birth.  3. The user enters another date of birth and clicks “Submit”.  4. The system verifies data and changes the date for the user.  5. The user can see a profile with another date. | |
| **Alternative scenarios and extensions:** | |

|  |  |
| --- | --- |
| **Title:** | User edit email, successful |
| ID: | FR-012 |
| **Description:**  The user wants to edit his information. He has an account, he goes to the profile page, where he sees all the information about the user. The user clicks “Edit” and sees an edit form with fields. After filling out the form, the user clicks “Submit” and sees the edited information about the user. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User sees his edited information. | |
| **Main scenario:**  1. The user wants to change his information and goes to the profile page, he selects "Edit".  2. The system shows username, email address, date of birth.  3. The user enters another email and clicks Submit.  4. The system checks the data and sends the activation to the new email address and an information message to the old email address.  5. The user must log in with a new email address.  6. The user can see the profile with a different email address. | |
| **Alternative scenarios and extensions:**  4. The system checks the data and sends the activation to the new email address and an information message to the old email address:  a. If the password is not strong enough, the system sends the message with password requirements.  b. Redirection to the field “Password”.  c. The user completes the field again. | |

|  |  |
| --- | --- |
| **Title:** | User edit password, successful |
| ID: | FR-013 |
| **Description:**  The user wants to edit his information. He has an account, he goes to the profile page where he sees all the information about the user. The user clicks “Edit” and selects “Change password”. After filing the form, the user clicks “Submit” and gets the email with a message about changing the password. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User gets the email with a message about changing the password. | |
| **Main scenario:**   1. The user wants to change his information and goes to the profile page, he selects “Edit”. 2. The system shows username, email address, date of birth. 3. The user selects “Change password”. 4. The user enters old password, new password, then confirms password and clicks Submit. 5. The system checks the data, changes the password and sends an information message to the email address. 6. 6. The user can see a mail that he changed his password. | |
| **Alternative scenarios and extensions:**  5. The system checks the data, changes the password and sends an information message to the email address:   1. If the email address is already in the database, the system sends the message “This email is already in use”. 2. Redirection to the field “Email”. 3. The user completes the field again. | |

|  |  |
| --- | --- |
| **Title:** | Creating a post open to everyone |
| ID: | FR-014 |
| **Description:**  The user wants to create a post that will be open to everyone. He is on the home page. He clicks to the field, enters the text and marks it as a public post. After that, post is saved to database and can be read by user followers on timeline or by every user on public timeline. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  The user posted a public post. | |
| **Main scenario:**  1. User log in  2. User is on a home page and clicks on the create new post button  3. User enters the text for the post to the text field  4. User marks post as public  5. User clicks "Post"  6. The system writes the post to the database  7. The post is shown in respective timeline | |
| **Alternative scenarios and extensions:**  5. The user clicks “Post”:  a. No connection with the server, the system sends a message “No connection”.  b. The post is not created and the user can not see it in the timeline | |

|  |  |
| --- | --- |
| **Title:** | User browses posts of users he follows |
| ID: | FR-015 |
| **Description:**  The user logs in and is on a home page. User browses posts created by other users he follows. He can scroll down through the newsfeed. Posts are presented from the newest one. When user reach the last post, newsfeed loads next, oldest posts. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  Timeline has generated latest posts for user | |
| **Main scenario:**   1. User login to the system 2. User goes to the post view page 3. User reads the posts and scroll down through the time line | |
| **Alternative scenarios and extensions:**   1. User reads the posts and scroll down through the time line 2. New batch of posts of is generated on timeline as soon as user reach the bottom of previous one 3. User reached end of the timeline (can be limited with timeframe), no new posts will be generated | |

|  |  |
| --- | --- |
| **Title:** | User browses posts of ALL userS |
| ID: | FR-016 |
| **Description:**  The user logs in and is on a home page. User browses posts of all users. He can scroll down through the newsfeed. Posts are presented from the newest one. When user reach the last post, newsfeed loads next, oldest posts. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  Timeline has generated latest posts for user | |
| **Main scenario:**   1. User login to the system 2. User goes to the discover page to browse public posts 3. User reads the posts and scroll down through the time line | |
| **Alternative scenarios and extensions:**  3) User reads the posts and scroll down through the time line   1. New batch of posts of is generated on timeline as soon as user reach the bottom of previous one 2. User reached end of the timeline (can be limited with timeframe), no new posts will be generated | |

|  |  |
| --- | --- |
| **Title:** | Unfollow option from home page |
| ID: | FR-017 |
| **Description:**  User is on the home page and sees its own timeline with posts he is following. User chooses other user to unfollow among posts visible on the timeline. User clicks "unfollow" button on one of posts. All posts of unfollowed user will disappear from the user timeline. User will no longer see posts of unfollowed user on his homepage. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User unfollowed other selected user. | |
| **Main scenario:**   1. User login to the system 2. User is on the home page 3. User click on unfollow button of another selected user from the posts timeline 4. Post creator is unfollowed, other user is removed from the followed users list 5. Posts of other user will not be/ will be showed on user's timeline | |
| **Alternative scenarios and extensions:**   1. User is on the home page 2. User is not following any users, user has no posts visible on home page, no posts to unfollow. 3. User browse posts and choose the post of user he wants to unfollow | |
|  | |

|  |  |
| --- | --- |
| **Title:** | follow option from discover page |
| ID: | FR-018 |
| **Description:**  User is on the discover page and sees its own timeline with posts he is following. User chooses other user to follow among posts visible on the timeline. User clicks "follow" button on one of posts. All posts of followed user will have their button text changed from “follow” to “unfollow”. User is now following this post author and this and other posts will be visible in the users timeline on the home page. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User followed other selected user. | |
| **Main scenario:**   1. User login to the system 2. User goes to the discover page 3. User clicks on follow button of another selected user from the posts timeline 4. Post creator is followed, other user is added to the followed users list 5. Posts of other user will be visible on user's timeline (home page) and button on all the posts of followed user change its comment to “unfollow”. | |
| **Alternative scenarios and extensions:** | |

|  |  |
| --- | --- |
| **Title:** | Unfollow option from discover page |
| ID: | FR-019 |
| **Description:**  User is on discover page and see its own timeline with posts he is following. User chooses other user to unfollow among posts visible on the timeline. User clicks "unfollow" button on one of posts. Button with unfollow sign will change to “follow” for all posts of unfollowed user. The user is no longer following the other user and will no longer see his posts on his own home page. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User unfollowed other selected user. | |
| **Main scenario:**   1. User login to the system 2. User goes to the discover 3. User clicks on unfollow button of another selected user from the posts timeline 4. Post creator is unfollowed, other user is removed from the followed users list 5. Posts that belongs to other user will have button with “follow” option on discover page. | |
| **Alternative scenarios and extensions:** | |

|  |  |
| --- | --- |
| **Title:** | User sees the list of all users he is following |
| ID: | FR-20 |
| **Description:**  User is on a home page. User wants to checks the list of all users he is following. User goes to the profile page. User clicks on the “Following” tab. The list with all users the user is following is showed. Each followed user is on separate line and has “unfollow” button. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User sees list of other users he is following. | |
| **Main scenario:**   1. User login to the system 2. User goes to the profile page. 3. User clicks on Following tab. 4. List with all followed users is showed to the user | |
| **Alternative scenarios and extensions:**  **3.** User clicks on Following tab.  **a**) User is not following any other user, user see information on how he can follow others. | |

|  |  |
| --- | --- |
| **Title:** | User unfollow other user from his followers list |
| ID: | FR-21 |
| **Description:**  User is on a home page. User wants to checks the list of all users he is following. User goes to the profile page. User clicks on the “Following” tab. The list with all users the user is following is showed. Each followed user is on separate line and has “unfollow” button. Users can unfollow other users by clicking on “unfollow” button placed by listed usernames. If user unfollows other user, the list is refreshed and unfollowed user is removed from the list. | |
| **The initial conditions:**  The user is registered.  The user is authorized. | |
| **Acceptance criteria:**  User sees list of other users he is following. | |
| **Main scenario:**   1. User login to the system 2. User goes to the profile page. 3. User clicks on Following tab. 4. List with all followed users is showed to the user. 5. User clicks “unfollow” button. 6. Unfollowed user is removed from the users followed list in the database. 7. Unfollowed user disappears from the list. | |
| **Alternative scenarios and extensions:**  **3.** User clicks on Following tab.  **a**) User is not following any other user, user see information on how he can follow others. | |

|  |  |
| --- | --- |
| **Title:** |  |
| ID: |  |
| **Description:** | |
| **The initial conditions:** | |
| **Acceptance criteria:** | |
| **Main scenario:** | |
| **Alternative scenarios and extensions:** | |