|  |  |
| --- | --- |
| Use case uc1: create profile |  |
| scope | Puppr application |
| level | User goal |
| primary actor | User |
| stakeholder and interests | * User: Wants an intuitive creation process, with working features of adding the wanted personal information. * Admin: Wants the profiles to be created correctly and according to the guidelines |
| preconditions | No other profile exists with the same email/username |
| success guarantee | A profile is created with username, password and it can further be managed and altered. All information is correctly added |
| main success scenario | 1. User selects to sign up under a new account. 2. Menu opens and the user fills the information needed (email, password, full name, etc.) 3. Information is accepted by the user and the system and the new user is created 4. New user info is stored in the database 5. User can now log in, access his profile, and interact with the feed whenever they want |
| Extensions (or alternative flows) | \*a. User with the same username already exists  1. Error message is shown  2. User can check and re-enter a new email or go directly to log in  2. No information provided in one or more fields   1. The system will show a warning notifying the user of the need to fill in the obligatory fields 2. The user can proceed to the next step after providing all the information needed to the system or exit and choose another option from log in page |
| Special Requirements | * Easy and user-friendly UI for account creation |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Often to very often |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc2 : manage profile |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants to be able to access and modify the private and public information in an easy and user-friendly way * Admin: Wants the users to have a certain degree of freedom in choosing the personal information they want to make public, while being able to manage their own account information |
| preconditions | A working account is already set up for the users |
| success guarantee | The selected profile data is modified and updated accordingly and immediately, so other users can see the new information in real time. |
| main success scenario | 1. User or admin log in with using their credentials 2. User or admin enters the “Manage profile” menu from “Settings” 3. User/Admin select the fields that they want to modify 4. The new data is entered and registered in the system 5. After all changes are done, the user can select to save or cancel them. 6. The “Manage profile” window is closed automatically, and the user/admin can proceed with any other activities, for example adding a post. |
| Extensions (or alternative flows) | 1a. Login information is incorrect   1. The user/admin will be prompted to check the information and try again 2. If the login succeeds, they can proceed to the next step, otherwise they can only see the main feed as a guest   2a. The new data entered by the user/admin is not valid   1. An error will be displayed, notifying the user about any limitations that need to be taken into account (eg: already existing username, any empty field, etc.) 2. After new data that conforms to the regulations of the system is entered, the error will disappear, and system changes will be saved   3a. User/Admin wants to add info about a new pet   1. The system will show a window will all the information the user can provide about the pet, including a picture 2. After adding all the needed information, they can save the changes 3. A new special part in their profile dedicated to the newly added pet will appear |
| Special Requirements |  |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Often |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc3: add post |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants a fast and intuitive way of adding new post and sharing media on the platform for other users to see. * Admin: Wishes for the system to function as a platform where both user types can easily share media |
| preconditions |  |
| success guarantee | Post will be added, saved by the system, and made available for all the users, admin, and guests to see, and for admins and users to also interact with. |
| main success scenario | 1. User/Admin selects the “Add post” option 2. User/Admin then can select from adding text, pictures, or a combination of both 3. After selecting what to post, they click the “Post” button 4. The post is added to the main wall with all the other posts, dated and made available for everyone to see and like or comment depending on their user type. |
| Extensions (or alternative flows) | 1. User not logged in    1. System will signal to the user the need to log on first and redirect them to that page    2. After successfully logging in, the user can go back to adding a post 2. User/Admin cancels the add post process    1. System will show a message asking for confirmation from the user       1. If user selects to proceed with not posting the content the post will be saved to drafts for easy later access and the user will be returned to the main page       2. If the user resumes the add process, they will be redirected to their post unfinished post. |
| Special Requirements | * Interactive and user-friendly UI for posting * Support of png and jpg formats |
| Technology and Data Variations List | 2a. For the future other formats might be wanted by the users, for example mp4 |
| Frequency of Occurrence | Could be almost continuous |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc4 : add comment |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants an easy and fast way of interacting with other users through commenting on posts * Admin: Wants to further ensure the social functionalities of the platform by allowing users to comment on each other's post and being able to leave their own comments |
| preconditions |  |
| success guarantee | Comment is added and saved to the list of comments for the certain post so everyone can see it and interact with it |
| main success scenario | 1. User/Admin clicks the comment area under a certain post 2. User/Admin types in the comment and clicks the “Add comment” button 3. After adding the comment, it is added to the list on comments for the certain post 4. User/Admin can continue adding more comments to that post, other posts or continue with other activities on the app |
| Extensions (or alternative flows) | 1a. User/Admin is not logged in   1. A pop-up window will require them to log in or sign up before being able to add a comment 2. After proceeding with the log in process, the use will be returned to the main flow window 3. They can return to their add comment activity or other activities   2a. “Add comment” is clicked without a comment being entered   1. Comment will not be added, and system will notify the user for the need to add text by changing the comment section color |
| Special Requirements | * Comment section updated at most 2 seconds after a new comment has been added * Support for English as the main system language |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Very often or nearly continuous. |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc5: add dog |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants to be able to share their dogs with the other users, this way contributing directly to the dog owner community * Admin: Wants the users to be able to showcase their dogs on their individual profile, while having the right to add their own dog information too. |
| preconditions |  |
| success guarantee | Dog is added to the owner’s profile and saved in the system. |
| main success scenario | 1. User/Admin picks the “Add Dog” option when managing their account. 2. User/Admin fills the name of the dog and can add a short description and a photo. 3. The “save” option is picked 4. New Dog information is saved and linked to the profile of the user or admin for everyone to see 5. The user/admin can continue with other activities or add another dog |
| Extensions (or alternative flows) | 1a. User/Admin is not logged in   1. A pop-up window will require them to log in or sign up before being able to manage an account and add a dog 2. After proceeding with the adding process, the use will be returned to the managing window where they can continue their activities   2a. “Save” is clicked without a name being entered for the dog   1. The system will notify the user/admin of the empty field and will save the new Dog information once the needed information is added   2b. User/Admin selects to exit before saving   1. The Dog information will not be saved and window exited |
| Special Requirements | * The system will refresh after a change has been committed in 0.5 seconds 90% of the time. |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Sometimes to often |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc6: LIKE POST |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants to be able to like other users’ or their own posts * Admin: Wants the users and admins to be able to like each other's post |
| preconditions |  |
| success guarantee | Number of likes will be increased, and new total number of likes will be shown with the related post. |
| main success scenario | 1. The user or admin selects a post from the wall 2. User/Admin clicks the “Like” button under the post 3. After clicking the button, number of likes will go up and updated number of likes will be shown with the post 4. User can continue interacting with the same post or continue with other activities on the app |
| Extensions (or alternative flows) | 1a. User/Admin is not logged in   1. A pop-up window will require them to log in or sign up before being able to like a post or continue as a guest    1. If they proceed with the login process, the user or admin will be returned to the main flow window where they can return to their activities and like the post    2. If they choose to continue as a guest, they will be only able to view the posts and profiles   2-3. Post is already liked   1. If the post has been already liked by the same user or admin, clicking the button will result in no change in the system |
| Special Requirements |  |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Could be nearly continuous. |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc7: LIKE comment |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants to be able to like other or their own comments * Admin: Wants all users and admins to be able to like each other's comments |
| preconditions |  |
| success guarantee | The number of likes will be increased and it will be shown together with the related comment |
| main success scenario | 1. User or admin selects a comment from the list related to a post 2. User/Admin clicks the “Like” button for the selected comment 3. After clicking the button, number of likes will go up and the updated number of likes will be shown with the comment 4. They can continue interacting with the same post or continue with other activities on the app |
| Extensions (or alternative flows) | 1a. User/Admin is not logged in   1. A pop-up window will require them to log in or sign up before being able to like a comment, or continue as a guest    1. If they proceed with the login/sign up process, the user or admin will be returned to the main flow window where they can return to their activities and like the comment    2. If they choose to continue as a guest, they will be only able to view the posts and profiles   2-3. Comment is already liked   1. If the comment has been already liked by the same user or admin, clicking the button will result in no change in the system |
| Special Requirements |  |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Could be very often |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc8: LIKE dog |  |
| scope | Puppr application |
| level | User and admin goal |
| primary actor | User, Admin |
| stakeholder and interests | * User: Wants to be able to like other or their own dogs * Admin: Wants all users and admins to be able to like each other's comments |
| preconditions |  |
| success guarantee | The number of likes will be increased and it will be shown together with the related dog object |
| main success scenario | 1. User or admin chooses to view a user profile 2. User/Admin clicks the “Like” button for a dog 3. After clicking the button, number of likes will go up and the updated number of likes will be shown with the dog post 4. They can continue interacting with the same user profile or continue with other activities on the app |
| Extensions (or alternative flows) | 1a. User/Admin is not logged in   1. A pop-up window will require them to log in or sign up before being able to like a dog post or continue as a guest    1. If they proceed with the login process, the user or admin will be returned to the main flow window where they can return to their activities and like the dog post    2. If they choose to continue as a guest, they will be only able to view the posts and profiles   2-3. Dog is already liked   1. If the post has been already liked by the same user or admin, clicking the button will result in no change in the system |
| Special Requirements |  |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Could be very often |
| Miscellaneous |  |

|  |  |
| --- | --- |
| **Use case uc9: Manage Posted content** |  |
| **scope** | Puppr application |
| **level** | Admin level |
| **primary actor** | Admin |
| **stakeholder and interests** | User: - Wants the platform and its posts to be well maintained and managed in order to have a pleasant experience while using it.  Admin: - Wants to be able to moderate or remove unwanted content to ensure user safety and experience |
| **preconditions** |  |
| **success guarantee** | The selected posts, comments or dogs will be edited or removed completely, changes saved, and system updated for all the users and admins. |
| **main success scenario** | 1. Admin logs in with the special credentials 2. System verifies and confirm the admin status 3. Main feed is shown in real time and delete functions are available for every post/comment. 4. An admin can select if they want to edit or delete a post or comment from the main feed or a dog’s information from a user profile 5. If the admin decides to proceed, the information will be edited or deleted accordingly from the main feed, profile, and the system’s memory 6. The admin can continue deleting more posts/comments or chose to log out and end their session. |
| **Extensions (or alternative flows)** | 1-2. Admin status not recognized   1. The Admin will be prompted by the system to check and re-enter the credentials 2. If the correct credentials are entered the system will be started in admin mode 3. If the admin fails to enter to correct login info, the system will not login or give any special admin roles to the user for security reasons   5-6. Admin decides not to proceed with deleting or editing the information   1. The selected information will not be affected or modified in any way 2. The admin will be returned to the main feed by the system 3. The admin can the continue his post managing activities or end his session |
| **Special Requirements** | * System updates the admin changes within 1 second 95% of the time. |
| **Technology and Data Variations List** |  |
| **Frequency of Occurrence** | Relatively often. |
| **Miscellaneous** | Open issues:   * Will a user be directly notified when their post has been deleted? * Should users be able to signal for comments or posts that ought to be moderated by the admin? * Can a user disagree with the admin’s choice to moderate their comment or post? |

|  |  |
| --- | --- |
| **Use case uc10: Manage users** |  |
| **scope** | Puppr application |
| **level** | Admin level |
| **primary actor** | Admin |
| **stakeholder and interests** | User: - Wants the platform to be a safe and friendly place for every user or viewer  Admin: - Wants to be able to moderate and limit the possibilities of users that have not conformed to the platform's rules. |
| **preconditions** |  |
| **success guarantee** | The selected user will be either blocked for an undetermined period of time, and so made unable to login with that account, or unblocked and made able to log in, post and comment freely again. |
| **main success scenario** | 1. Admin logs on and gets identified by the system 2. Admin selects the user’s profile who has been not conforming to the platform regulations or a user previously blocked 3. The admin can select the “Block user” or, respectively, the “Unblock user” option 4. After the Admin proceeds with the option the user’s capabilities of logging in the app with those credentials will be removed or returned accordingly |
| **Extensions (or alternative flows)** | 1a. Admin status not recognized   1. The Admin will be prompted by the system to check and re-enter the credentials 2. If the correct credentials are entered the system will be started in admin mode 3. If the admin fails to enter to correct login info, the system will not login or give any special admin roles to the user for security reasons |
| **Special Requirements** | * System updates the admin changes and notifies the user within 2 seconds 95% of the time. |
| **Technology and Data Variations List** |  |
| **Frequency of Occurrence** | Rarely |
| **Miscellaneous** | Open issues:   * Will an admin be able to block another admin? |

|  |  |
| --- | --- |
| Use case uc11: manage Dog |  |
| scope | Puppr application |
| level | User goal |
| primary actor | User |
| stakeholder and interests | * User: Wants to have the ability to remove or edit the information related to their dogs * Admin: Wants the users to have the freedom to modify the information about the dogs they own |
| preconditions | User has already a set up account |
| success guarantee | The selected dog data is modified and updated accordingly and immediately, so other users can see the new information in real time. |
| main success scenario | 1. User logs in using their credentials (username and password) 2. User opens their profile page and decides to delete or edit the information related to one of their dogs 3. User clicks the related “delete” or “edit” buttons 4. The data will be deleted or edited accordingly 5. Changes will be saved, and profile page updated across the app |
| Extensions (or alternative flows) | 1a. Login information is incorrect   1. The user/admin will be prompted to check the information and try again 2. If the login succeeds, they can proceed to the next step, otherwise they can only see the main feed as a guest   2a. No dog information is linked to the user’s profile   1. The system will not display any edit or delete buttons until the user completes an “add dog” operation in the app   3a. User selects the delete option   1. The dog data will be deleted immediately, and profile page updated   3b. User selects the edit option   1. A new window showing the current information related to the respective dog will pop up 2. The user can select what to change and what to leave the same way    1. User chooses to save the changes:       * The new old information will be replaced with the new one and changes committed    2. User chooses to exit the edit menu:       * The old information will remain unedited |
| Special Requirements |  |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Rarely |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc12: manage post |  |
| scope | Puppr application |
| level | User goal |
| primary actor | User |
| stakeholder and interests | * User: Wants to have the ability to remove or edit their posts after posting them to the app * Admin: Wants the users to have the possibility to fix any mistakes or delete any of their posts freely |
| preconditions | User has already a set up account |
| success guarantee | The selected post data is modified and updated accordingly and immediately, so other users can see the new information in real time. |
| main success scenario | 1. User logs in using their credentials (username and password) 2. User opens their profile page and decides to delete or edit of one their posts 3. User clicks the related “delete” or “edit” buttons for one of the posts 4. The data is deleted or edited accordingly if the changes are saved 5. All changes are committed and updated for all users |
| Extensions (or alternative flows) | 1a. Login information is incorrect   1. The user/admin will be prompted to check the information and try again 2. If the login succeeds, they can proceed to the next step, otherwise they can only see the main feed as a guest   2a. User has no posts   1. The system will not display any edit or delete buttons until the user completes an “add post” operation in the app   3a. User selects the delete option   1. The post data will be deleted immediately, as well as the comments and likes linked to it 2. Both the profile page and the main feed will be updated for everyone   3b. User selects the edit option   1. A new window showing the current information related to the respective post will pop up 2. The user can select what to change and what to leave the same way    1. User chooses to save the changes:       * The new old information will be replaced with the new one and changes committed    2. User chooses to exit the edit menu:       * The old information will remain unedited |
| Special Requirements | System updates the admin changes and notifies the user within 2 seconds 95% of the time. |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Often |
| Miscellaneous |  |

|  |  |
| --- | --- |
| Use case uc13: manage comment |  |
| scope | Puppr application |
| level | User goal |
| primary actor | User |
| stakeholder and interests | * User: Wants to have the ability to remove or edit their comments after posting them to on the app * Admin: Wants the users to have the possibility to fix any mistakes or delete any of their comments freely |
| preconditions | User has already a set up account |
| success guarantee | The selected comment data is modified and updated accordingly and immediately, so other users can see the new information in real time. |
| main success scenario | 1. User logs in using their credentials (username and password) 2. User opens a post’s page and decides to delete or edit of one their comments made on the said post 3. User clicks the related “delete” or “edit” buttons for one of the posts 4. The data is deleted or edited accordingly if the changes are saved 5. All changes are committed and updated for all users |
| Extensions (or alternative flows) | 1a. Login information is incorrect   1. The user/admin will be prompted to check the information and try again 2. If the login succeeds, they can proceed to the next step, otherwise they can only see the main feed as a guest   2a. User has no comments on the post   1. The system will not display any edit or delete buttons to any other comment until the user adds a comment to the specific post   3a. User selects the delete option   1. The comment will be deleted immediately 2. Both the profile page and the main feed will be updated for everyone   3b. User selects the edit option   1. A new window showing the current information related to the respective comment will pop up 2. The user can select what to change and what to leave the same way    1. User chooses to save the changes:       * The new old information will be replaced with the new one and changes committed    2. User chooses to exit the edit menu:       * The old information will remain unedited |
| Special Requirements | System updates the admin changes and notifies the user within 2 seconds 95% of the time. |
| Technology and Data Variations List |  |
| Frequency of Occurrence | Rarely |
| Miscellaneous |  |