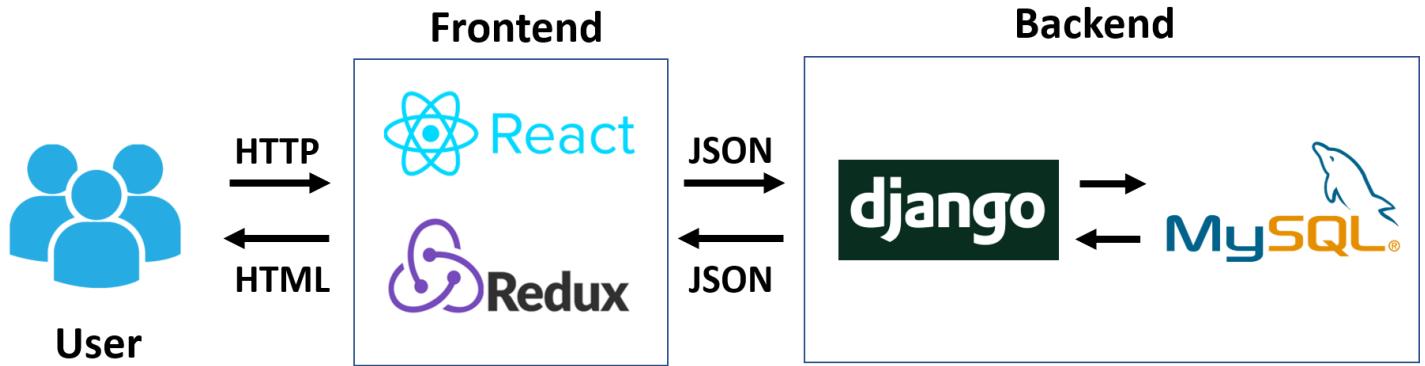


design_planning.md

System Architecture

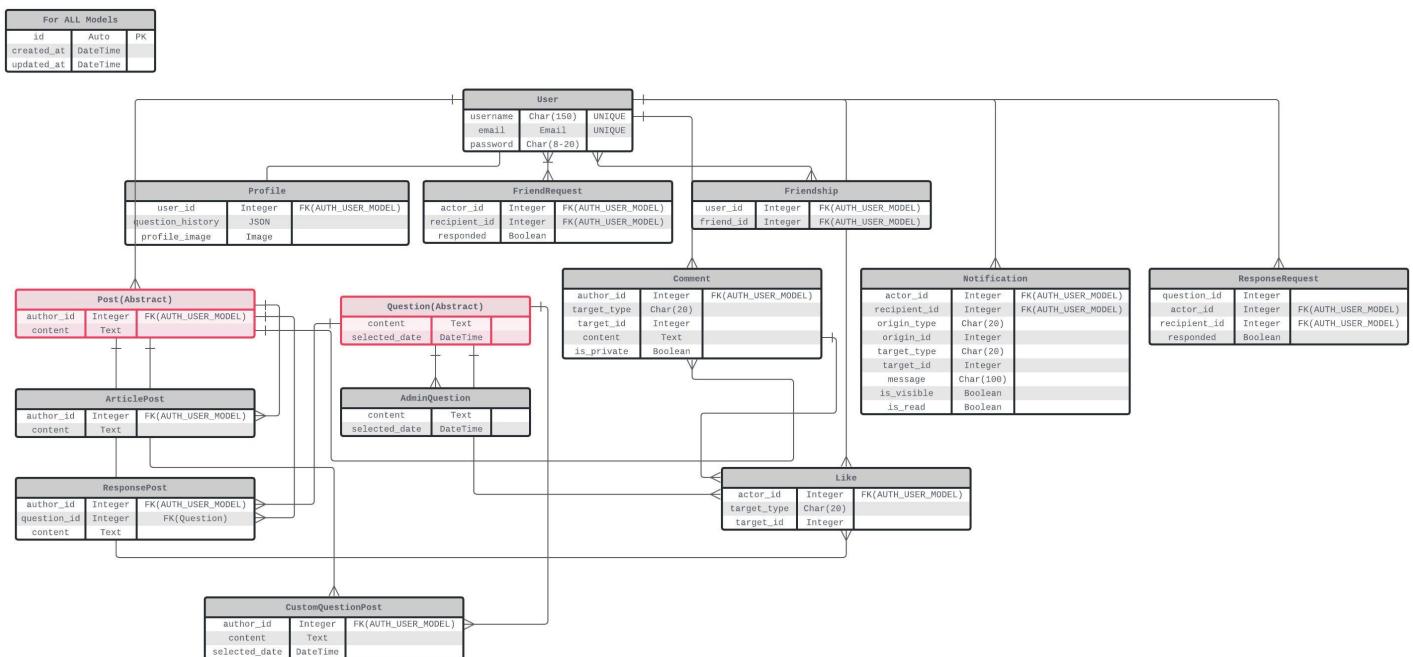


- **Frontend**
 - React.js: frontend library
 - Redux: state management
- **Backend**
 - Django
 - MySQL: RDB

Backend

Model ER-Diagram

[Model ER-Diagram] [Google Drive Link](#)



Backend API

Model	API	GET	POST	PUT	DELETE
User	/api/user/signup/	X	Create new user	X	X

Model	API	GET	POST	PUT	DELETE
	/api/user/select-questions/	Get questions for sign up	Submit selected questions at sign up	X	X
	/api/user/login/	X	Sign user in	X	X
	/api/user/logout/	Sign out	X	X	X
	/api/user/:id/	Get specified user	X	X	X
X	/api/user/search/:query/	Get result of username search	X	X	X
(Friendship)	/api/user/:id/friends/	Get friend list for specified user	X	X	X
	/api/user/friendship/:friend_id/	Get friendship status with specified user	Create new friendship with specified user	X	Delete friendship with specified user
(FriendRequest)	/api/user/friend-requests/:recipient_id/	Get specified friend request	Create new friend request from actor to recipient	X	X
Question	/api/feed/questions/:id/	Get specified question	X	X	X
	/api/feed/questions/daily/	Get daily question list	X	X	X
	/api/feed/questions/daily/:user_id/	Get daily question list for specified user	X	X	X
ResponseRequest	/api/feed/questions/request-response/	X	Create new response request	X	X
Feed	/api/feed/all/	Get all feed list	X	X	X
	/api/feed/friend/	Get friends feed	X	X	X
	/api/feed/anonymous/	Get anonymous feed	X	X	X
	/api/feed/:user_id/	Get feed for specified user	X	X	X
CustomQuestion	/api/feed/custom-questions/	X	Create new custom question	X	X
	/api/feed/custom-questions/:id/	Get specified custom question	X	Edit custom question	Delete custom question
Article	/api/feed/articles/	X	Create new article	X	X

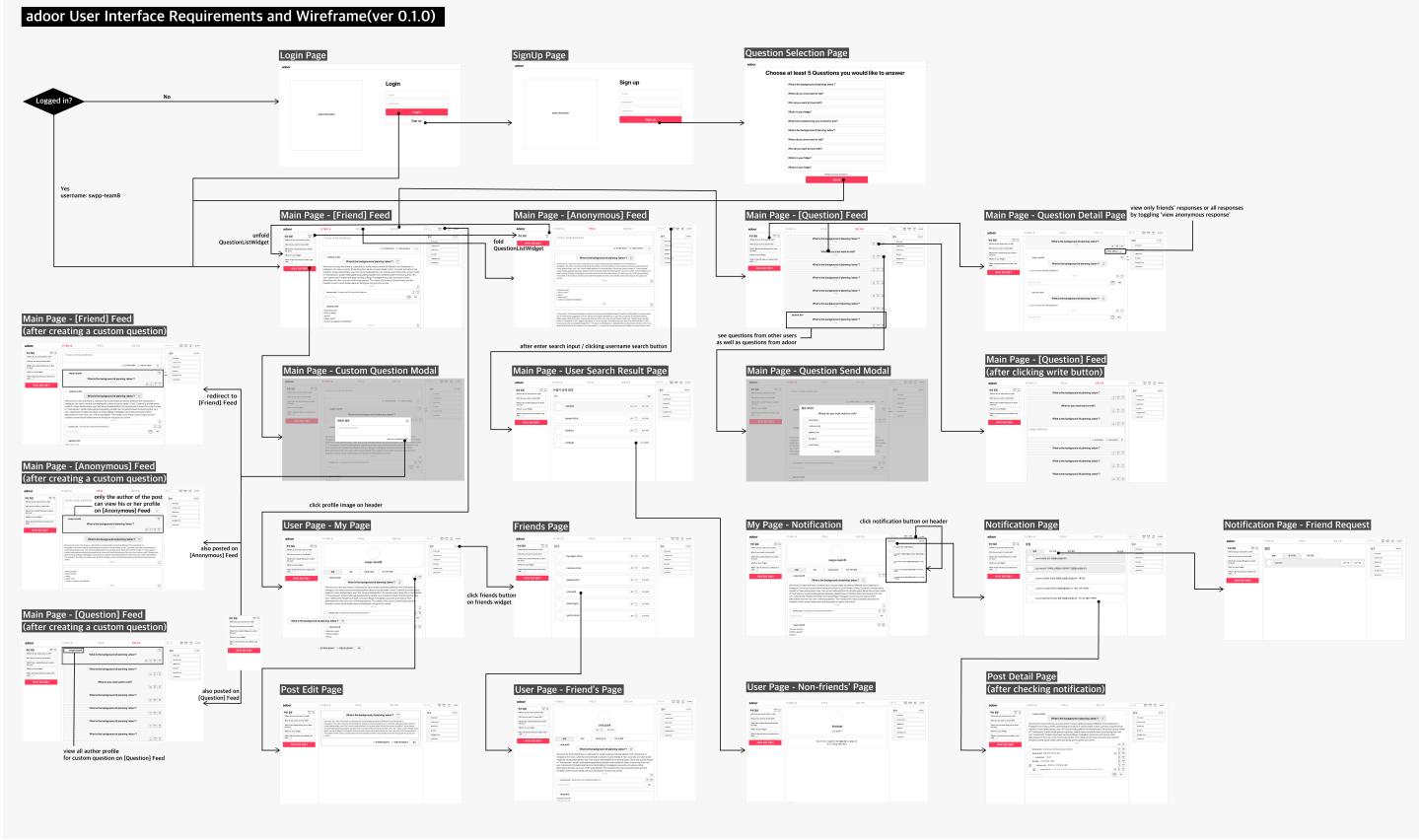
Model	API	GET	POST	PUT	DELETE
	/api/feed/articles/:id/	Get specified article	X	Edit specified article	Delete specified article
Response	/api/feed/responses/	X	Create new response	X	X
	/api/feed/responses/:id/	Get specified response	X	Edit specified response	Delete specified response
Comment	/api/comments/	X	Create new comment	X	X
	/api/comments/:id/	X	X	X	Delete specified comment
Like	/api/likes/	X	Create new like	X	X
	/api/likes/:id/	X	X	X	Delete specified like
Notification	/api/notifications/	Get notification list	X	Mark all notifications as read	X
	/api/notifications/:id/	X	X	Mark specified notification as read	X
	/api/notifications/friend-requests/	Get notification list on friend requests	X	X	X
	/api/notifications/friend-requests/:recipient_id/	X	X	X	Delete specified notification on friend request
	/api/notifications/response-requests/	Get notification list on response requests	X	X	X

Frontend

UI Requirements(ver 0.1.0 updated from ver 0.0.2)

[Wireframe] Google Drive Link

[All Pages] Google Drive Link



Major update

- remove the function of sharing settings(friends/anonymous) for custom question
- add with send button in response post
- replace with custom question button
- change My Page and Friend Profile Page to User Page
- replace friend-related buttons with

Minor update

- add key color
- add like count for post author
- replace edit/delete button of post with more-dropdown
- add private-comment-button
- add share seetings(friends/anonymous) in Post Edit Page

Frontend Components

Pages

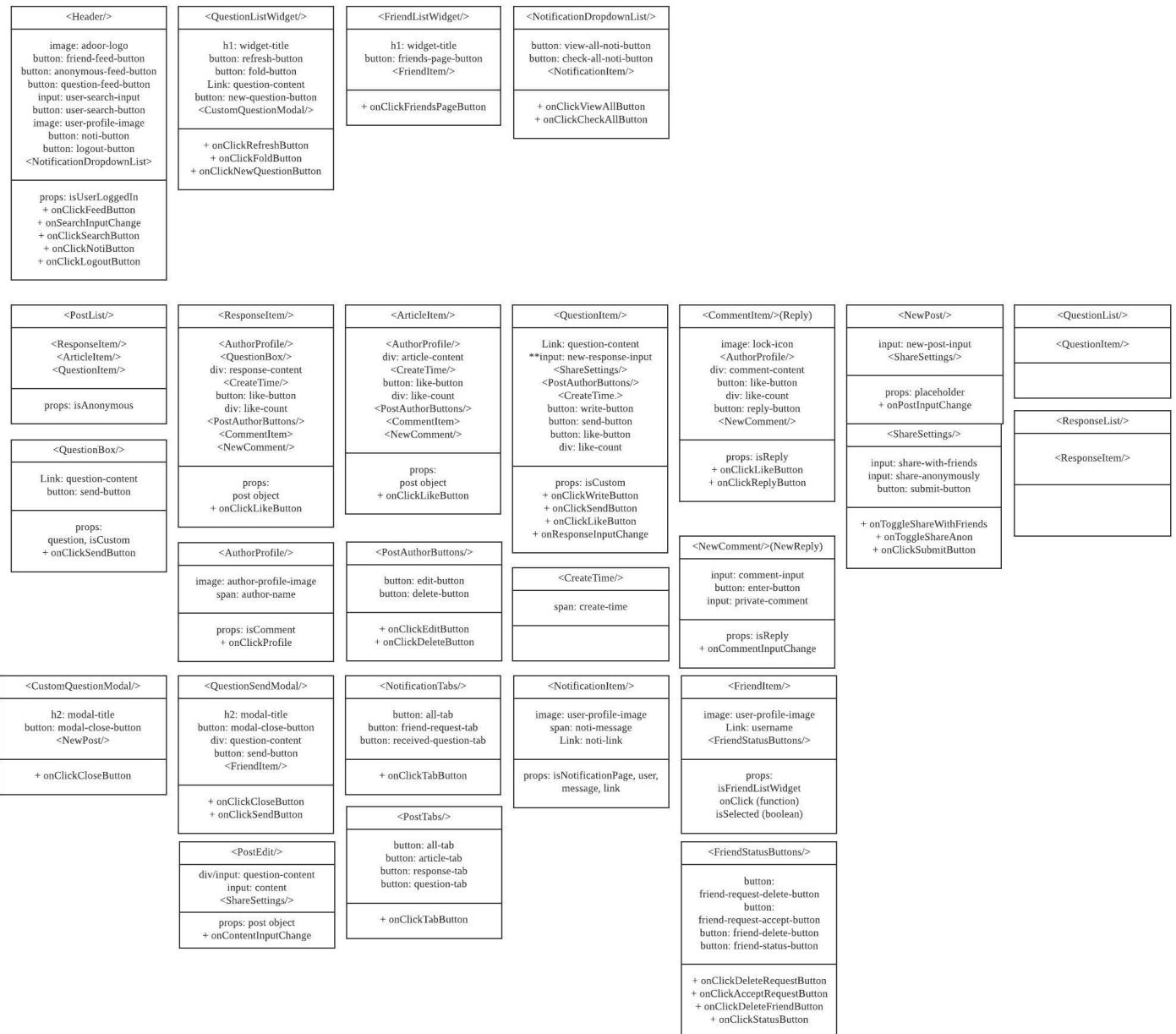
[Frontend Components - Pages] Google Drive Link

Page												
<div style="display: flex; align-items: center; justify-content: space-between;"> { All Pages Common <Header/> } </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 10px; vertical-align: top;"> Login <i>(/login)</i> <p>h1: login-title input: email-input input: password-input button: signup-button button: submit-button image: adoor-illustration</p> <p>+ onClickSignupButton + onClickSubmitButton + onInputChange</p> </td> <td style="padding: 10px; vertical-align: top;"> SignUp <i>(/signup)</i> <p>h1: signup-title input: username-input input: email-input input: password-input button: submit-button button: login-button image: adoor-illustration button: check-username-button</p> <p>+ onClickLoginButton + onClickSubmitButton + onCheckUsername + onInputChange</p> </td> <td style="padding: 10px; vertical-align: top;"> QuestionSelection <i>(/select-questions)</i> <p>h1: question-selection-title div: question-choice button: submit-button</p> <p>+ onQuestionSelect + onClickSubmitButton</p> </td> </tr> </table>	Login <i>(/login)</i> <p>h1: login-title input: email-input input: password-input button: signup-button button: submit-button image: adoor-illustration</p> <p>+ onClickSignupButton + onClickSubmitButton + onInputChange</p>	SignUp <i>(/signup)</i> <p>h1: signup-title input: username-input input: email-input input: password-input button: submit-button button: login-button image: adoor-illustration button: check-username-button</p> <p>+ onClickLoginButton + onClickSubmitButton + onCheckUsername + onInputChange</p>	QuestionSelection <i>(/select-questions)</i> <p>h1: question-selection-title div: question-choice button: submit-button</p> <p>+ onQuestionSelect + onClickSubmitButton</p>									
Login <i>(/login)</i> <p>h1: login-title input: email-input input: password-input button: signup-button button: submit-button image: adoor-illustration</p> <p>+ onClickSignupButton + onClickSubmitButton + onInputChange</p>	SignUp <i>(/signup)</i> <p>h1: signup-title input: username-input input: email-input input: password-input button: submit-button button: login-button image: adoor-illustration button: check-username-button</p> <p>+ onClickLoginButton + onClickSubmitButton + onCheckUsername + onInputChange</p>	QuestionSelection <i>(/select-questions)</i> <p>h1: question-selection-title div: question-choice button: submit-button</p> <p>+ onQuestionSelect + onClickSubmitButton</p>										
<div style="display: flex; align-items: center; justify-content: space-between;"> { All Pages Common (without Login, SignUp, QuestionSelection Page) <QuestionListWidget/> <FriendListWidget/> } </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 10px; vertical-align: top;"> FriendFeed <i>(/friends)</i> <p><NewPost/> <PostList/></p> </td> <td style="padding: 10px; vertical-align: top;"> AnonymousFeed <i>(/anonymous)</i> <p><NewPost/> <PostList/></p> </td> <td style="padding: 10px; vertical-align: top;"> QuestionFeed <i>(/questions)</i> <p><QuestionList/> <QuestionSendModal/></p> </td> <td style="padding: 10px; vertical-align: top;"> QuestionDetail <i>(/questions/:id)</i> <p><QuestionItem/> <ResponseList/> input: view-anonymous-response-toggle</p> <p>+ onToggleAnonymousSwitch</p> </td> </tr> <tr> <td style="padding: 10px; vertical-align: top;"> UserPage <i>(/users/:id)</i> <p>div: user-profile-background image: user-profile-image span: username <FriendStatusButtons/> <PostTabs/> <PostList/></p> </td> <td style="padding: 10px; vertical-align: top;"> UserSearchResultPage <i>(/users/search?q=')</i> <p>h1: title <FriendItem/></p> </td> <td style="padding: 10px; vertical-align: top;"> FriendsPage <i>(/users/:id/friends)</i> <p>h1: title <FriendItem/></p> </td> <td></td> </tr> <tr> <td style="padding: 10px; vertical-align: top;"> NotificationPage <i>(/notifications)</i> <p>h1: title <NotificationTabs/> <NotificationItem/> <FriendItem/></p> </td> <td style="padding: 10px; vertical-align: top;"> PostEditPage <i>(/post/:id/edit)</i> <p><PostEdit/></p> </td> <td style="padding: 10px; vertical-align: top;"> PostDetailPage <i>(/post/:id)</i> <p><ResponseItem/> <ArticleItem/> <QuestionItem/></p> </td> <td></td> </tr> </table>	FriendFeed <i>(/friends)</i> <p><NewPost/> <PostList/></p>	AnonymousFeed <i>(/anonymous)</i> <p><NewPost/> <PostList/></p>	QuestionFeed <i>(/questions)</i> <p><QuestionList/> <QuestionSendModal/></p>	QuestionDetail <i>(/questions/:id)</i> <p><QuestionItem/> <ResponseList/> input: view-anonymous-response-toggle</p> <p>+ onToggleAnonymousSwitch</p>	UserPage <i>(/users/:id)</i> <p>div: user-profile-background image: user-profile-image span: username <FriendStatusButtons/> <PostTabs/> <PostList/></p>	UserSearchResultPage <i>(/users/search?q=')</i> <p>h1: title <FriendItem/></p>	FriendsPage <i>(/users/:id/friends)</i> <p>h1: title <FriendItem/></p>		NotificationPage <i>(/notifications)</i> <p>h1: title <NotificationTabs/> <NotificationItem/> <FriendItem/></p>	PostEditPage <i>(/post/:id/edit)</i> <p><PostEdit/></p>	PostDetailPage <i>(/post/:id)</i> <p><ResponseItem/> <ArticleItem/> <QuestionItem/></p>	
FriendFeed <i>(/friends)</i> <p><NewPost/> <PostList/></p>	AnonymousFeed <i>(/anonymous)</i> <p><NewPost/> <PostList/></p>	QuestionFeed <i>(/questions)</i> <p><QuestionList/> <QuestionSendModal/></p>	QuestionDetail <i>(/questions/:id)</i> <p><QuestionItem/> <ResponseList/> input: view-anonymous-response-toggle</p> <p>+ onToggleAnonymousSwitch</p>									
UserPage <i>(/users/:id)</i> <p>div: user-profile-background image: user-profile-image span: username <FriendStatusButtons/> <PostTabs/> <PostList/></p>	UserSearchResultPage <i>(/users/search?q=')</i> <p>h1: title <FriendItem/></p>	FriendsPage <i>(/users/:id/friends)</i> <p>h1: title <FriendItem/></p>										
NotificationPage <i>(/notifications)</i> <p>h1: title <NotificationTabs/> <NotificationItem/> <FriendItem/></p>	PostEditPage <i>(/post/:id/edit)</i> <p><PostEdit/></p>	PostDetailPage <i>(/post/:id)</i> <p><ResponseItem/> <ArticleItem/> <QuestionItem/></p>										

Components

[Frontend Components - Components] Google Drive Link

Component



Frontend Description

All pages excluding [Authorization Related] pages described below include Question List Widget, Friend List Widget and Header which includes a logo of adoor, user search bar, the user's profile icon, notification icon and logout button by default.

(<QuestionListWidget/> , <FriendListWidget/> , <Header/>).

Page

[Authorization Related]

1. Login Page (/login)

- Let a user login with his/her own account.
 - If the user clicks the 'Sign up' button, redirect to the Sign Up Page.

2. Sign Up Page (/signup)

- Let a user create a new account.
 - If the user clicks the 'Sign up' button, redirect to the Question Selection Page.

3. Question Selection Page (/select-questions)

- Show 10 questions selected by adoor.
- Let users choose 5 questions that he/she would like to answer.
- If the user clicks the 'Done' button, create a new account and redirect to the Friend Feed Page.

[Post Related]

4. Friend Feed Page (/friends)

- Show all posts written by the authorized user and his/her friends. (<PostList/>)
- Let users create a new post by setting an audience for his/her post. (<NewPost/>)
- If the user clicks a certain post, redirect to the Post Detail Page.
- If the user clicks the 'Submit' button, add the new post to the feed([Friend/Anonymous]) according to the audience he/she set right before.
- Components included: <PostList/> , <NewPost/>

5. Anonymous Feed Page (/anonymous)

- Show all posts written by the authorized users that are shared anonymously. (<PostList/>)
- Let users create a new post by setting an audience for his/her post. (<NewPost/>)
- If the user clicks a post, redirect to the Post Detail Page.
- If the user clicks the 'Submit' button, add the new post to the feed([Friend/Anonymous]) according to the audience he/she set right before.
- Components included: <PostList/> , <NewPost/>

6. Post Detail Page (/post/:id)

- Show a detailed page of a post including comments and replies. (<ResponseItem/> , <ArticleItem/> , <QuestionItem/>)
- Let users create a new comment or a new reply for the post.
- If the user clicks the 'Edit' button for his/her own post, redirect to the Post Edit Page.
- If the user clicks the 'Delete' button for his/her own post, delete the post.
- Components included: <ResponseItem/> , <ArticleItem/> , <QuestionItem/>

7. Post Edit Page (/post/:id/edit)

- Let users edit his/her own post. (<PostEdit/>)
- If the user clicks the 'Submit' button, update the post.
- Components included: <PostEdit/>

[Question Related]

8. Question Feed Page (/questions)

- Show all posted questions. (<QuestionList/>)
- If the user clicks the 'Send' button, open the Question Send Modal. (<QuestionSendModal/>)
- If the user clicks a question, redirect to the Question Detail Page.
- Components included: <QuestionList/> , <QuestionSendModal/>

9. Question Detail Page (/questions/:id)

- Show a detailed page of a question. (<QuestionItem/>)
- Show all responses to the question. (<ResponseList/>)
- Components included: <QuestionItem/> , <ResponseList/>

[User Management Related]

10. User Page (/users/:id)

- Show all posts the user wrote. (<PostList/>)
- Let users select the tab([All/Article/Response/Question]). (<PostTabs/>)
- Components included: <PostList/> , <PostTabs/>

11. User Search Result Page (/users/search?q=")

- Show all users the user searched for. (<FriendItem/>)
- If the user clicks one of the users' profile icon or name, redirect to the User Page of the selected user.
- Components included: <FriendItem/>

12. Friends Page (/users/:id/friends)

- Show all friends of the user. (<FriendItem/>)
- Let users manage his/her own friends.
- If the user clicks one of his/her friend's name or profile icon, redirect to the User Page of the selected friend.
- Components included: <FriendItem/>

13. Notification Page (/notifications)

- Show all notifications the user received. (<NotificationItem/>)
- Let users select the tab([All/Friend Request/Received Questions]). (<NotificationTabs/>)
- If the user selects [Friends Request] tab, show all friends who sent requests. (<FriendItem/>)
- Let users accept or decline the friends request(s).
- If the user selects [Received Questions] tab, show all questions he/she received.
- Components included: <NotificationItem/> , <NotificationTabs/> , <FriendItem/>

Widget

1. Question List Widget

- Show 5 random daily questions recommended by adoor.
- If the user clicks the 'Refresh' button, regenerate questions randomly.
- If the user clicks the 'Fold/Unfold' button, fold/unfold the widget.
- If the user clicks a question, redirect to the Question Detail Page.

2. Friend List Widget

- Show all friends of the user. (<FriendItem/>)
- If the user clicks the friend's username or profile icon, redirect to the User Page of the selected friend.
- If the user clicks the 'Manage' button, redirect to the Friends Page.

Modal

1. Custom Question Modal

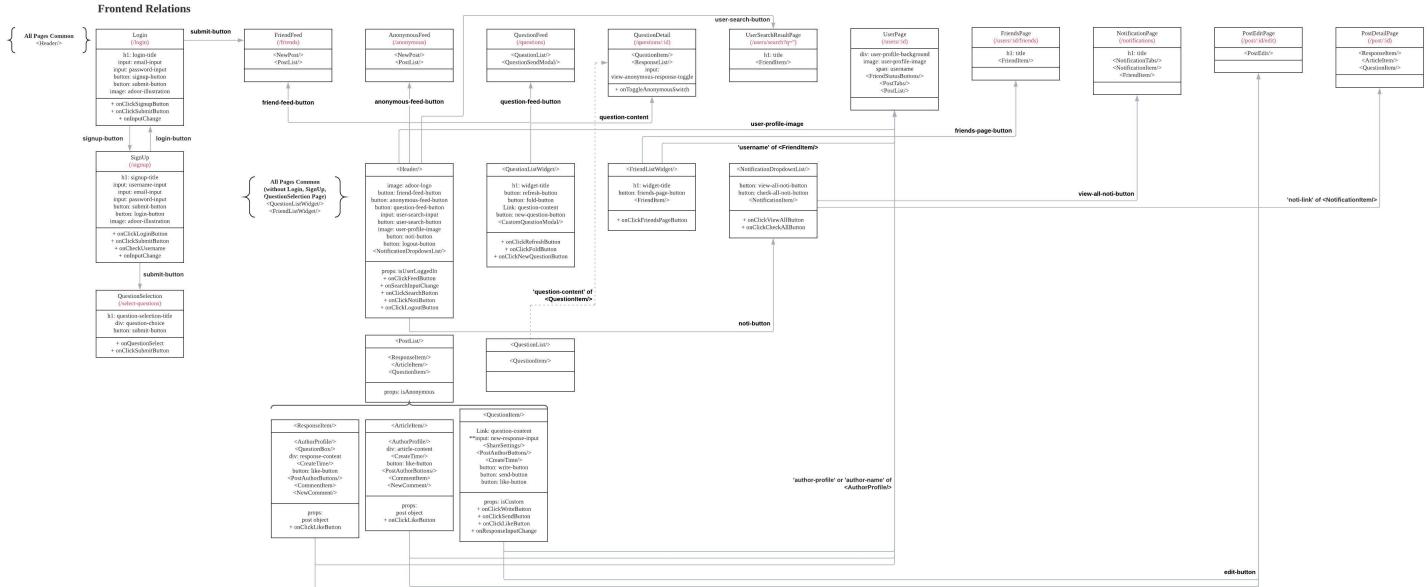
- Let users create a question by setting an audience for his/her post. (<NewPost/>)
- If the user clicks the 'Submit' button, redirect to the feed ([Friend/Anonymous]) where the question is posted.
- If the user clicks the 'Close' button, close the modal.

2. Question Send Modal

- Let users send a question to his/her friend(s).
- Let users choose friends to whom to send. (<FriendItem/>)
- If the user clicks the 'Send' button, send the question to selected friends.
- If the user clicks the 'Close' button, close the modal.

Frontend Components Relationship

[Frontend Components Relationship] Google Drive Link



States of Reducer

Reducer	State	Description
User	user	contains user information
Questions	sampleQuestions	array of sample questions to pick @ sign up page
	randomQuestions	random questions (updated when refresh button clicked)
	recommendedQuestions	recommended questions for user
	selectedQuestion	question object @ question detail page
	selectedQuestionResponses	response objects of selected question
	dailyQuestions	daily questions displayed @ question feed
Posts	anonymousPosts	array of anonymous post objects
	friendPosts	array of friends' post objects
	selectedUserPosts	array of posts of currently visited user
	selectedPost	post object at post detail page
Friends	friendList	array of friend objects with profile
	selectedUser	profile objects of currently visited user page
	searchUserResults	results of user search with username
Notifications	receivedNotifications	array of all received notifications

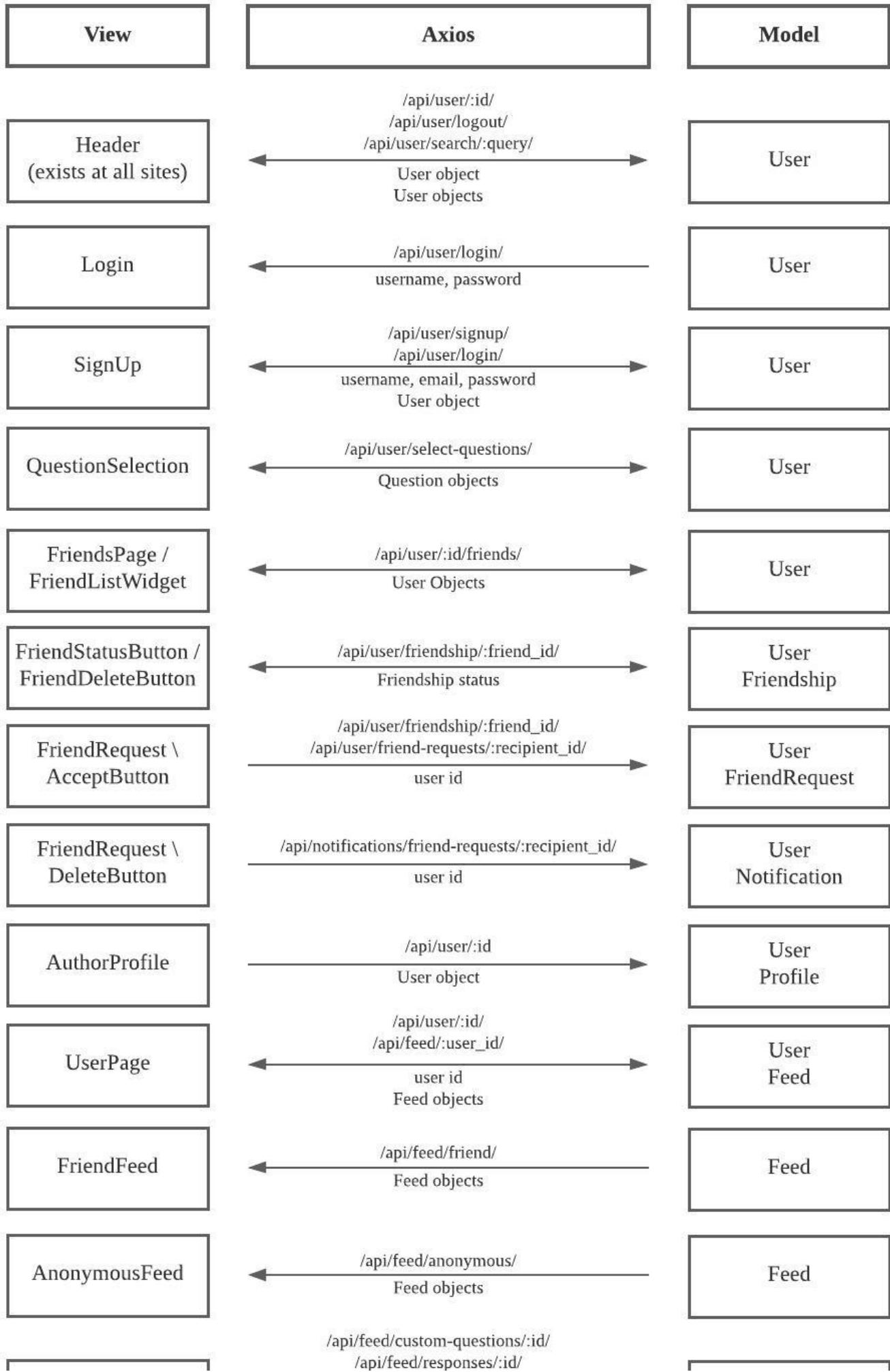
Actions of Reducer

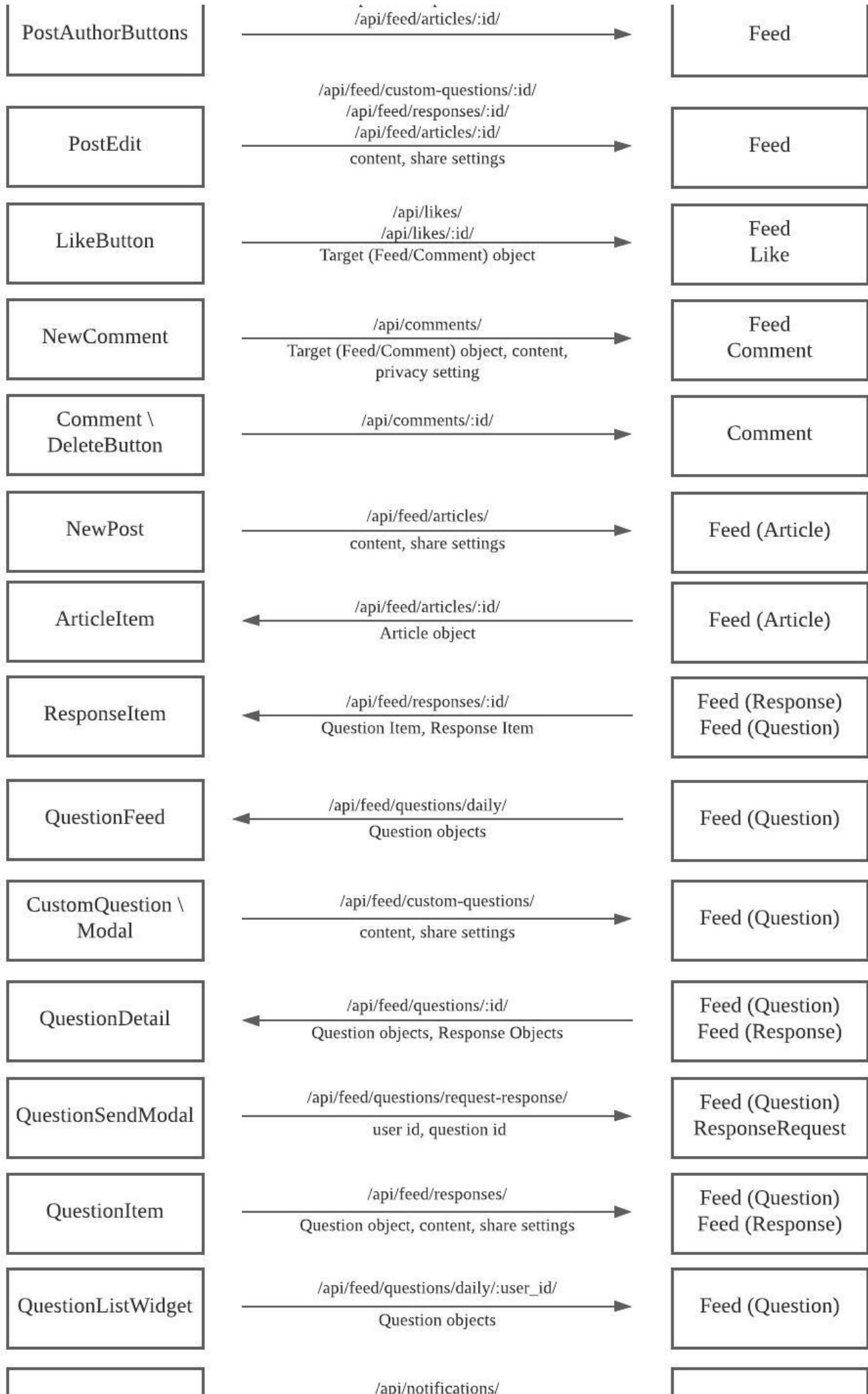
Reducer	Action	Function
User	login({ email, password })	send email/password, and receive current user profile
	logout()	logout current user
	signUp({ email, username, password, questions })	send user info(email, username, password, favorite questions), check validation, and login
Questions	getRecommendedQuestions(userId)	get recommended questions for current user
	getDailyQuestions()	get all daily questions for Question feed

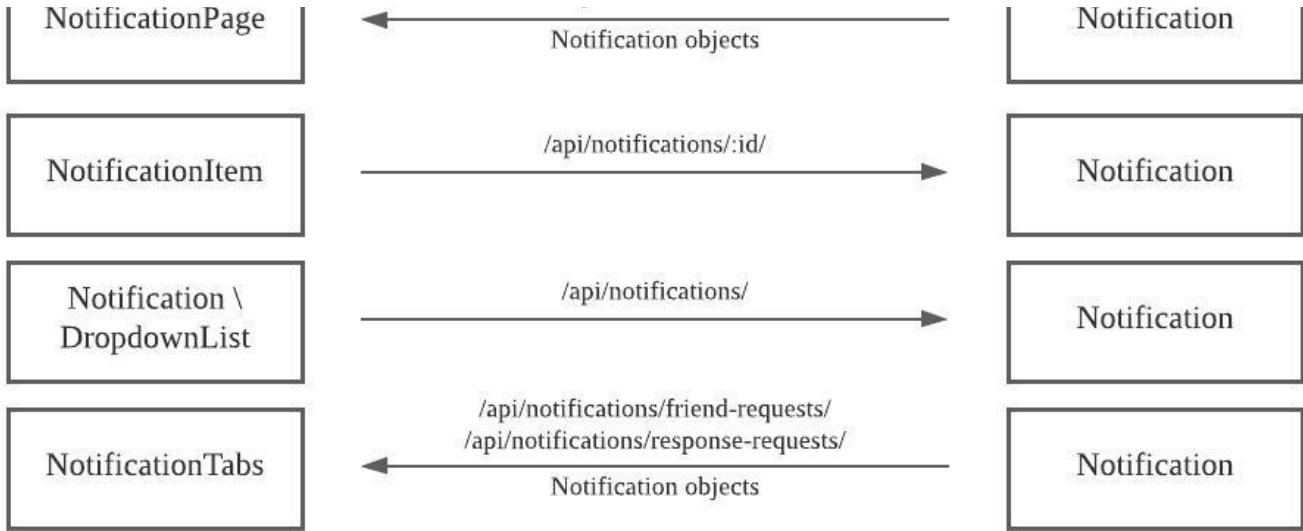
Reducer	Action	Function
	getRandomQuestions()	get random questions when refresh button on Question List Widget is clicked
	createCustomQuestion(questionObj)	create a custom question
	createResponseRequests(receiverId, questionId)	send response request to selected friends on selected question
	getSelectedQuestion(questionId)	get selected question for Question Detail Page
	getResponsesByQuestion(type, questionId)	get responses by selected questions with type[friend/anonymous], questionId
	getSampleQuestions()	get sample questions at signup
Posts	CreateResponse(postObj)	create a response
	CreateArticle(postObj)	create a article
	getPostsByType(type[anon/friends/user], userId)	get posts(responses, articles, questions) with type[friend/anonymous/user] (and userId for user type)
	resetPosts()	reset all posts state when page changes
	editMyPost(postId, newPostObj)	edit selected post
	deleteMyPost(postId)	delete selected post
	getSelectedPost(postId)	get selected post for Post Detail Page
	toggleLike(type, id)	toggle like for selected post/comment/reply
	createComment(commentObj, postId)	create a comment for selected post
	deleteComment(commentId)	delete selected post
	createReply(commentObj, commentId)	create a reply for selected comment
	deleteReply(commentId)	delete selected reply
	getMyFriendList()	get all friends(userId, username)
	addFriend(friendId)	add friend to friend list when accept friend request
	deleteFriend(friendId)	delete friend from friend list when delete friend request
	searchUserByUsername(query)	get username search result
	sendFriendRequest(userId)	send friend request when click Friend Request Button
	getNotifications()	get all notifications
	readNotification(notificationId)	read selected notification
	readAllNotifications()	read all notifications
	deleteNotification(notificationId)	delete selected notification when delete + accept friend request

Axios Design

[Axios Design] Google Drive Link





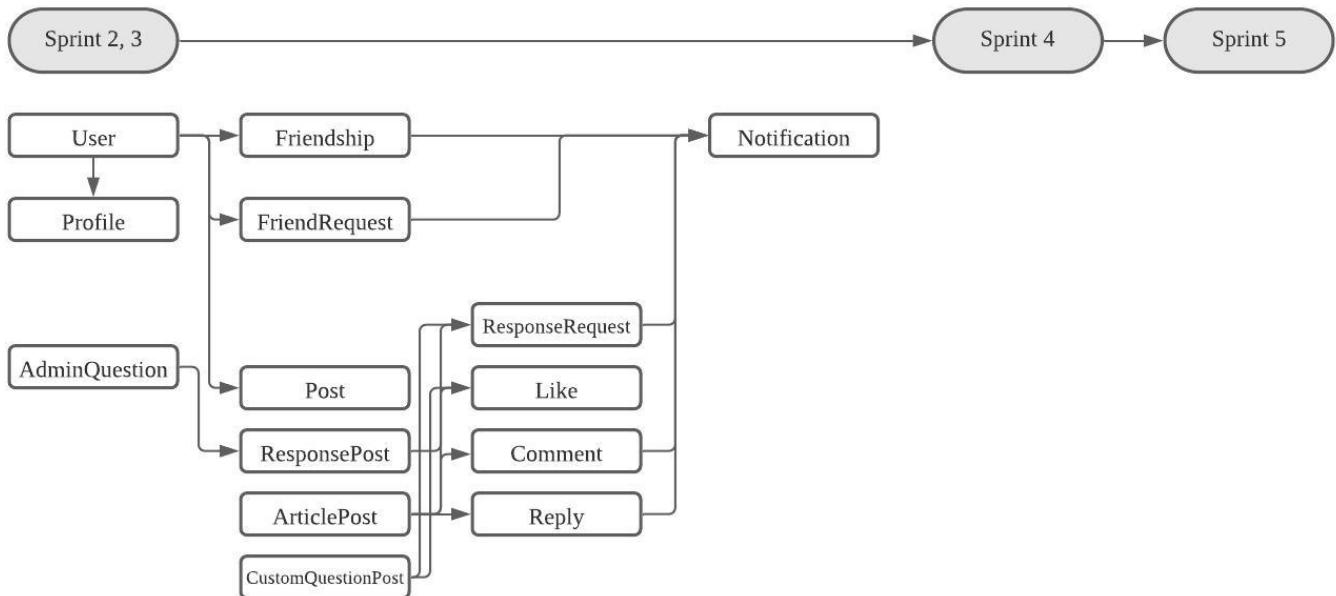


Implementation Plan

Project Development Dependencies

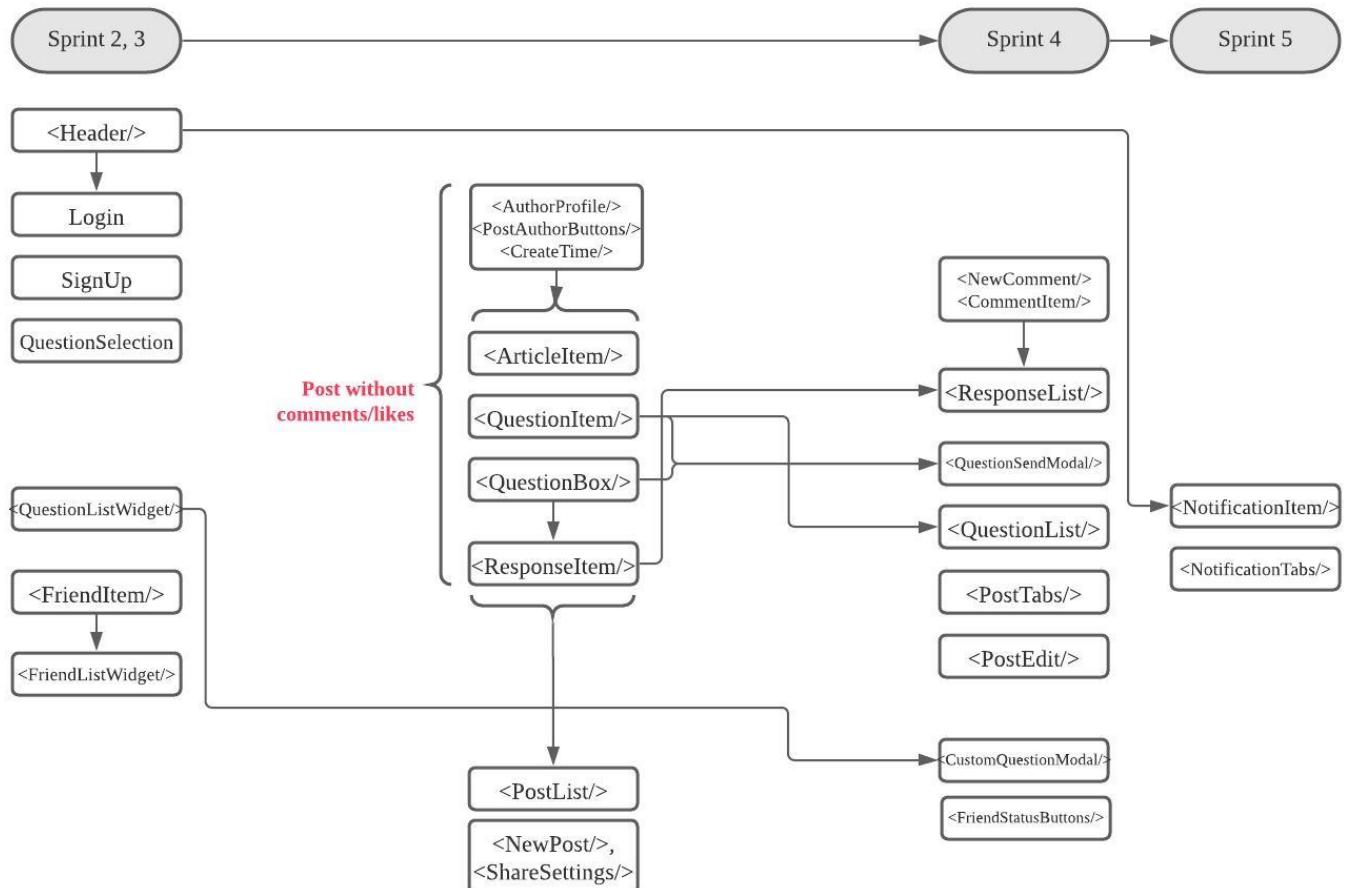
[Project Development Dependencies] Google Drive Link

<Backend Dependencies>



ML: Question Recommendation

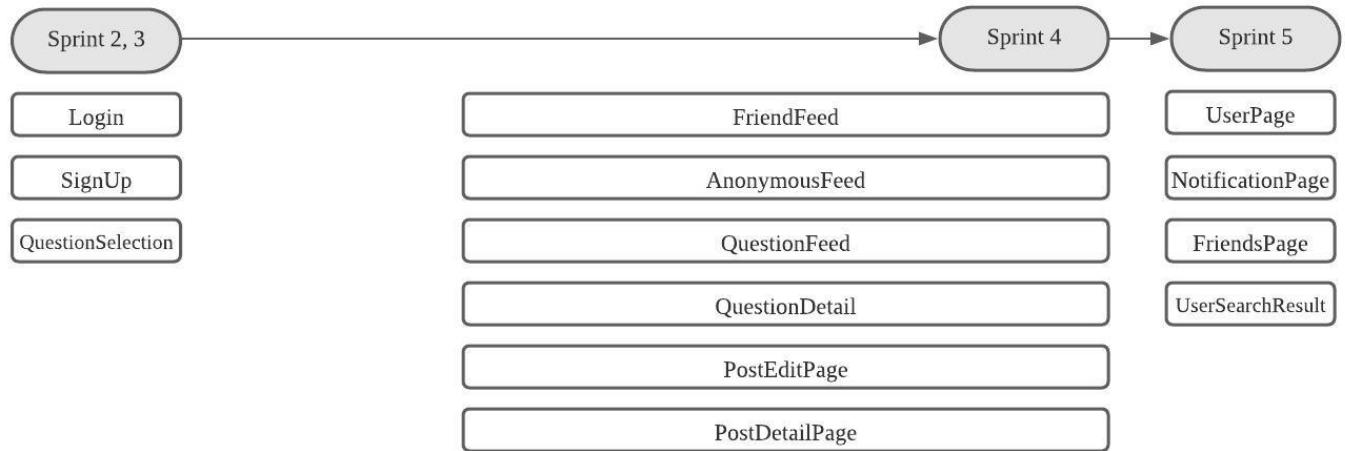
<Frontend Dependencies>



MainPage Layout

UserPage Layout

NotificationPage Layout

~~tags~~

Feature-task difficulties & sprint

Page	Feature	Tasks	Priority	Time(days)	Sprint	Assignees	Finished
Authentication	Sign Up	Write User Information	Medium	1	2	규리	
Provide Sample Questions Data	Medium	1	2	규리			
Show Sample Questions	Medium	1	2	규리			
Pick Questions	Medium	1	2	규리			
Login	User Log In	Medium	1	2	지나		
Logout	User Log Out	Medium	1	2	지나		
Common		Pagination (Inifinite Scroll)	Medium	2	4		
Post / Comment timestamp	Medium	1	4				
Header	Implement App Header	Medium	1	2	진선		
QuestionListWidget	Import Questions	High	?	2-4			
Implement QuestionListWidget	Medium	1	2	진선			
Schedule Questions	High	?	2-4				
Implement Question Recommendation Algorithm	High	?	3-5				
View Recommended Questions	High	?	2-5				
Refresh & Get Random Questions	Medium	2	2-3				
Fold / Unfold Widget	Medium	1	2				
Create New Question	Medium	2	4				
Go To Question Detail Page When Clicking On Question	Medium	1	3-4				

Page	Feature	Tasks	Priority	Time(days)	Sprint	Assignees	Finished
FriendListWidget	View Friends List	High	2	2-3			
Implement FriendListWidget	Medium	1	2	진선			
Visit Friend's Page By Clicking On Profile	Medium	1	5				
Main Page	[Friend, Anonymous, Question] Feed	Browse Answers By Friends	High	2	3-4		
Browse Answers By Anonymous	Medium	2	3-4				
View Question Feed	Medium	2	3-4				
Send Questions	Medium	2	4				
Write Answers to Selected Question	High	2	3-4				
Create Post	Write Article	High	1	3			
Set Visibility of New Post	Medium	1	3				
User Feed	User Posts	View Posts of Page Owner	High	2	4		
View Profile of Page Owner	High	2	4				
Post Detail		View Question And Answer	High	1	4		
View Comments	Medium	1	4				
Like Post	Medium	1	4				
Write Comment	Medium	1	4				
Write Comment Reply	Low	2	4				
Visit Author's Page By Clicking Icon	Medium	1	4				
Edit Post	Medium	1	3-4				
Delete Post	Medium	1	3				
Question Detail		View Current Question	High	1	3-4		
View Answers of Current Question	High	1	3-4				
Show Friend Modal When Send Button Clicked	Medium	1	4				
Send Current Question To Selected Friend	Medium	2	4				
Show Input Area When Write Button Clicked	Medium	1	3				
Post Answer to Current Question	Medium	1	3				

Page	Feature	Tasks	Priority	Time(days)	Sprint	Assignees	Finished
My Page	Profile	View Profile Info	Low	1	5		
Friends	Manage Friends	Low	2	5			
Search Friend By Username	Medium	1	5				
Add Friend	High	1	4-5				
Notification		View Friends Request	High	1	5		
View Response Request	Medium	2	5				
Reactions To My Posts	Medium	1	4				
Answers To Questions that I requested to friend	Low	2	5				
Click Friend Request => Go To Noti Page (Friend Request Tab Selected)	Low	1	5				
Click Question Request => Go To Noti Page (Question Request Tab Selected)	Low	1	5				
Click Reaction To My Post => Go To Post Detail Page	Low	1	5				
Click Answers To Questions I Requested => Go To Post Detail Page	Low	1	5				
Notification Page		View All Notifications	High	2	3-5		
View All Friends Request	Medium	1	3-5				
Respond to Friend Request	Medium	1	3-5				
View All Received Questions	Medium	1	3-5				

Testing Plan

In each sprint, every components and modules will be tested, with code coverage over 80%. The following frameworks will be used for testing:

- React: Enzyme & Jest
- Django: Python unit test

We will also leverage E2E testing(functional testing) to test that the whole system is working properly, from client-side to server-side. This will be done in the final sprint using the following frameworks:

- React : Cypress
- Django : Python test

Also, in sprint 4, we will start to use Integration testing to test that desired changes occur in the UI when events are fired. All required properties of a particular component should properly pass to the descendant component. These are the frameworks to be utilized:

- React & Redux : jest, enzyme
- Django : Python test