

Use Cases (done by Paul, Xin and Geneva)

1.

Use Case Name: BrowseQuestions

Participating Actors: User

Goal: View question titles from the main page in the list from along with the number of answers they have

Trigger: User wishes to browse questions and navigates to the main page of the app

Preconditions: User knows how to open the app and how to go to the main page from other activities

Postconditions: On success display questions with number of answers

Basic Flow:

1. System displays a list of questions with the number of answers it has for browsing

Exceptions:

User Stories: 1, 14

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 1 (As a user, I want browse questions.)

User Story 14(As a user, I want to see how many answers a question has received)

Open Issues:

Acceptance tests: This use case is tested by first making a question list, creating a question and adding it to the question list (because if there are no questions in the question list, there is nothing to browse). After, the question is obtained from the question list and answers are added to the question. Then, the question is re-inserted back into the question list. The answers a question receives is then checked.

2.

Use Case Name: ViewQuestionsAndAnswers

Participating Actors: User

Goal: View the answers related to a question

Trigger: User chooses the question he/she wish to view

Preconditions: User knows how to open the app and how to go to the main page from other activities

Postconditions: On success display Q&A view of the question selected

Basic Flow:

1. System displays a list of Questions asked by users for browsing.
2. User selects a question
3. System brought up a new view displaying only the question selected and answers to the question in chronological order

Exceptions:

3. There are no answers to the question selected
- 3.1 System displays “No answers for this question yet” in the answer section

User Stories: 2

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 2 (As a user, I want to view a question and its answers.)

User Story 22(As a user, by default, I should see the most fresh comments)

Open Issues:

Acceptance tests: To test this use case, a question is created and an answer is added to this question (because in order to view answers for a question, there must be a question and at least one answer). Exception 3 is tested by checking if there are answers to a question (which is done before an answer is added to the question), and

since there are no answers, the string “No answers for this question yet” is displayed to the user.

3.

Use Case Name: ViewReplies

Participating Actors: User

Goal: View replies to a question or an answer.

Trigger: User chooses on a question to answer.

Preconditions: User knows which question or answer they want to view the replies to.

Postconditions: System displays the list of replies to the question/answer that the user long clicked on.

Basic Flow:

1. System displays the list of replies to the corresponding question/answer indicated.

Exceptions:

1. There are no replies
 - 1.1 Display “No Replies” message.

User Stories: 3

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 3 (As a user, I want to view the replies to a question or answer)

Open Issues:

Acceptance tests: To test this use case, a question is created and a reply is added to the question. Then a question list is created and the question added to the question list. An answer is then created and a reply added to this answer. Then the answer is added to the question. Then the number of replies to the question and to the answer are checked and then the question and answer are checked to see if they contain the replies added to them. Exception 1 is checked by checking the question and answer to see if they have any replies (which is done in the Junit tests before replies are added to

a question or answer) and since there are no replies, the string “No replies” is displayed to the user.

4.

Use Case Name: AskQuestion

Participating Actors: Author, SysadminquestionName2 is empty

Goal: Author adds a question to the list.

Trigger: Author chooses the AskAQuestion option

Preconditions: Author knows the question they want to ask & (optional) the photo they want to attach.

Postconditions: On success, the question is added & success message appears.

Basic Flow:

1. System prompts Author to enter a question & optionally add a photo.
2. Author submits his/her question & (optional) attaches a photo.
3. System displays the question & photo and wait for confirmation
4. Author submits the question
5. System returns to the previous/ with the recently added question at the top of the list.

Exceptions:

4. Question entered is whitespace characters only.
 - 4.1 System prompts author to re-enter a valid question
5. Not connected to the network.
 - 5.1 System stores the question locally.
 - 5.2 System pushes to the network the next time it is connected to the network.

User Stories: 4

Qualities:

Constraints:

Includes: AddPhotoFromGallery (Use Case 7)

Extends:

Related Artifacts:

Notes:

User Story 4 (As an author, I want to make questions)

User Story 7 (As an author, I want to attach a picture to my questions or my answers)

Open Issues:

Acceptance tests: A question list is first created and a question is created. A picture is then created and added to the question and the question is then added to the question list. To test Exception 4, a string called questionName2 with only whitespaces is created (intended to simulate a empty question), and then the String method trim() is applied to it. Since the string length is 0, the user is prompted to re-enter a valid question. To test Exception 5, the network connectivity of the question list is obtained and since the connectivity is false, "No network connectivity, push online later" is returned.

5. AddAnswer

Use Case Name: AddAnswer

Participating Actors: Author, Sysadmin

Goal: Author adds an answer to a question.

Trigger: Author chooses the AddAnAnswer option.

Preconditions: Author knows the answer they want to submit and (optional) the photo they want to attach.

Postconditions: On success, the answer is added to the question & a success message appears.

Basic Flow:

1. System prompts Author to enter an answer & optionally add a photo.
2. Author submits his/her answer & (optional) attaches a photo.
3. System displays the answer & photo and wait for confirmation
4. User confirms their Answer.
5. System returns to the previous/Q&A screen with the recently added answer at the top of the list of answers.

Exceptions:

4. Answer entered is whitespace characters only.
 - 4.1 System prompts author to reenter an answer via a toast message
5. Not connected to the network.
 - 5.1 System stores the question locally.
 - 5.2 System pushes to the network the next time it is connected to the network.

Qualities:

Constraints:

Includes: AddPhotoFromGallery (Use Case 7)

Extends:

Related Artifacts:

Notes:

User Story 5 (As an author, I want to answer questions by making an answer)

User Story 7 (As an author, I want to attach a picture to my questions or my answers)

Open Issues:

Acceptance tests: To test this use case, a question list and a question are created. The question is then added to the question list. The question is obtained from the question list (to simulate answering a question) and then an answer is then added to the question and then the question is re-added to the question list. The question is then checked to see if it has the answer. For Exception 4, a String with only whitespaces is created (to simulate an empty or only whitespace character answer) and then the String method trim() is applied to it. Since the string is empty, the user is prompted to re-enter a valid answer. To test Exception 5, the network connectivity of the question list is obtained and since the connectivity is false (by default for testing purposes), "No network connectivity, push online later" is returned.

6.

Use Case Name: AddReply

Participating Actors: Author

Goal: Author adds a reply to a question or answer.

Trigger: Author chooses the AddReply option.

Preconditions: Author knows the reply/comment they want to submit and to which item they want to reply to (question or answer).

Postconditions: On success, the reply is added to the question or answer.

Basic Flow:

1. System prompts author to enter their reply.
2. Author submits his/her reply.
3. System displays the reply and wait for confirmation
4. Author confirms the reply.
5. System returns to the previous screen with the most recently added reply at the top of the list of replies for that question or answer.

Exceptions:

4. Reply is empty or whitespace characters only
 - 4.1. System prompts author to reenter a valid reply
5. Not connected to the network.
 - 5.1 System stores the question locally.

5.2 System pushes to the network the next time it is connected to the network.

Qualities:

Constraints:

Includes:

Extends: ViewReplies (Use Case 3)

Related Artifacts:

Notes:

User Story 6 (As an author, I want to reply to questions and answers to clarify things)

Open Issues:

Acceptance tests: In order to test this use case, a question list is created and a question is created. Then a reply is added to the question. Then the question is checked to see if there is a reply for it. Then an answer is created (without errors) and a reply is created and added to the answer. Finally, the answer is added to the question and the question is added to the question list. To test Exception 4, a String with only whitespaces called emptyReply is created (to simulate an empty or only whitespace character answer) and then the String method trim() is applied to it. Since the string is empty, display a prompt to re-enter a valid reply To test Exception 5, the network connectivity of the question list is obtained and since the connectivity is false (by default, since this is a mock Junit test), "No network connectivity, push online later" is returned.

7.

Use Case Name: AddPhotoFromCamera

Participating Actors: Author, Sysadmin

Goal: Attach a photo to a question or answer

Trigger: Author chooses the AddPhotoFromCamera option.

Preconditions: Author knows how to take a picture.

Postconditions: On success, photo is attached to the question or answer.

Basic Flow:

1. System prompts author to take a photo.
2. Author takes a photo.
3. Sysadmin verifies that the photo is not larger than 64kb.
4. System returns to previous screen with preview of the image.

Exceptions:

3. Photo is larger than 64kb.

3.1 System displays “image is too large” message & prompts author to select another photo.

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts: AskAQuestion (Use Case 1), AddAnAnswer (Use Case 2), AddPhotoFromGallery(Use Case 7)

Notes:

User Story 7 (As an author, I want to attach a picture to my questions or my answers)

Open Issues:

Acceptance Tests: (1) Try adding a photo with size $\leq 64\text{kb}$ to a question and check that that picture is successfully attached to the question (the question's picture is not null). (2) Try adding photo with size $\leq 64\text{kb}$ to an answer and check that the picture is successfully attached to the answer (the answer's picture is not null). (3) Try a picture whose size is $>64\text{kb}$ and ensure that it is classified as too large.

8.

Use Case Name: AddPhotoFromGallery

Participating Actors: Author, Sysadmin

Goal: Attach a photo to a question or answer

Trigger: Author chooses the AddPhotoFromGallery option.

Preconditions: Author knows how to take a picture.

Postconditions: On success, photo is attached to the question or answer.

Basic Flow:

1. System prompts author to select a photo.
2. Author selects a photo.
3. Sysadmin verifies that the photo is not larger than 64kb.
4. System returns to previous screen with preview of the image.

Exceptions:

3. Photo is larger than 64kb.

3.1 System displays “image is too large” message & prompts author to

select another photo.

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts: AskAQuestion (Use Case 1), AddAnAnswer (Use Case 2), AddPhotoFromCamera (Use Case 6)

Notes:

User Story 7 (As an author, I want to attach a picture to my questions or my answers)

User Story 8 (As a sysadmin, I do not want the pictures to be large (>64kb))

Open Issues:

Acceptance Tests: (1) Try adding a photo with size $\leq 64\text{kb}$ to a question and check that that picture is successfully attached to the question (the question's picture is not null). (2) Try adding photo with size $\leq 64\text{kb}$ to an answer and check that the picture is successfully attached to the answer (the answer's picture is not null). (3) Try a picture whose size is $>64\text{kb}$ and ensure that it is classified as too large.

9.

Use Case Name: PhotoSizeLimit

Participating Actors: Sysadmin

Goal: Limits the attached photo size to be smaller than 64 kb

Trigger: Author/User attached a photo greater than 64kb

Preconditions: Author/User have photo in their gallery or knows how to take a photo

Postconditions: On success, photo size greater than 64kb is rejected.

Basic Flow:

1. System prompts author to select a photo.
2. Author selects a photo size greater than 64kb.
3. Sysadmin verifies that the photo is not larger than 64kb.
4. System displays "image is too large" message & prompts author to select another photo.

Exceptions:

3. Photo is smaller than 64kb.
 - 3.1 System will not complain

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts: AskAQuestion (Use Case 1), AddAnAnswer (Use Case 2), AddPhotoFromCamera (Use Case 6)

Notes:

User Story 8 (As a sysadmin, I do not want the pictures to be large (>64kb))

Open Issues:

Acceptance Tests: Create a question, then (1) try a picture that is >64kb and check that the system complains & deems it too large. (2) Try a picture that is <=64kb and ensure that the system has no complaints, essentially confirming that the picture is <=64kb and thus a valid picture, and then the picture is successfully attached to the question.

10.

Use Case Name: SortQuestionBy

Participating Actors: User

Goal: Display a sorted list of questions.

Trigger: User chooses the SortBy menu option

Preconditions: User knows how they want the questions to be sorted by (date - most recent / least recent, picture - with / without)

Postconditions: On success, display the sorted list of questions.

Basic Flow:

1. System prompts user to select method to sort by.
2. User selects the method they would like.
3. System displays the list of questions in sorted order in the method chosen.

Exceptions:

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 9 (As a user, I want to sort questions by if they have pictures)

User Story 10 (As a user, I want to sort questions by date or some scoring system)

Open Issues:

Acceptance Tests: Make a question and set the date to be 5 seconds before the current date(time), then make another question and set that date to 3 seconds before the current date(time), (and hence more recent than the first question), then make a third question with the date set to be the current date(time). Then we set the upvotes of each question to be 1, 10, and 5 respectively. Then we add the questions to a question list. We create the appropriate questionlist that are in the correct order for each of the following : most recent, least recent, most upvoted and least up voted, then call each of the sortBy methods on the list and ensure that the list matches the corresponding correct list that we made.

11.

Use Case Name: Upvote

Participating Actors: User

Goal: Selected Question/Answer upvoted

Trigger: User chooses to upvote a question/answer

Preconditions:at least one question and/or answer need to exist

Postconditions: On success, the number of upvote beside a question/answer needs to increment by one

Basic Flow:

1. User selects the question/answer they want to upvote
2. System displays the incremented upvoted number beside the question/answer

Exceptions:

1. User already upvoted the same question/answer
 - 1.1 the upvoted number will not increment.
 - 1.2 A message will display notifying the user they have already upvoted the question/answer

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 11(As a user, I want to upvote the question of other users)

User Story 12 (As a user, I want to upvote the answers of other users)

Open Issues:

Acceptance Tests: (1) To test that a user cannot upvote the same question/answer more than once, we first create a user and set the username. Then we make a question and upvote that question; and we add that question to a list of questions that this user/username has up voted, and check that the question was successfully upvote. Then we do the same for an answer. Then we check to see if that question/answer has already been upvoted by that user and ensure that it has been to show that the question/answer was successfully up voted and the system knows that the user has already upvoted it and cannot do so again.

12.

Use Case Name: SeeMostUpvoted

Participating Actors: User

Goal: Display the most upvoted question and answer

Trigger: User browses a question

Preconditions: There are at least one question/answer with upvote

Postconditions: On success, display the most upvoted question/answer

Basic Flow:

1. User browses question
2. System displays the most upvoted question on the top of the question list
3. User browses the answers to a question
4. System displays the most upvoted answer at the top of the answer list

Exceptions:

1. there are no questions
 - 1.1 System displays “no questions currently” message
2. there are no upvoted questions
 - 2 .1 System displays questions without a most upvoted question
3. there are no answers to a question
 - 3.1 System displays “no answers currently” message
4. There are no upvoted answers
 - 4.1 systems displays answers in chronological order.

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 13 As a user, I want to see how many answers a question has received)

Open Issues:

Acceptance tests: To test SeeMostUpvoted, a user is created and a new question list is created. Then, exception 1 is tested by having the system check if the question list is empty and if it returns 'no questions currently'. Then, a new question is created and added to the question list. The system then tests exception 2 and checks if there are any upvoted questions. Since the question is not currently upvoted, the system displays the questions without a most upvoted question. Since there are no answers added yet to the question, exception 3 is tested by checking the answer list length and since it is zero, returns no answers currently. Afterwards, two new answers are created without any upvotes and added to test exception 4 and since no answers are upvoted, the getUpVote function will return 0 and system should display answers in chronological order. The upvotes of two answers are then set to 5 and 10 and system checks to see if answer 2 is selected as the most upvoted answer.

13.

Use Case Name: Search

Participating Actors: User

Goal: Return search results for a given query

Trigger: User chooses the search option

Preconditions: User knows what they want to search for

Postconditions: On success, system displays results for the given search query.

Basic Flow:

1. System prompts user to enter their query.
2. User enters a query.
3. System displays a list of question or answers relating to the given query.

Exceptions:

3. If there are no matching results.
 - 3.1 System displays a “no matches” message
 - 3.2 System returns to step 1

User Stories: 15

Additional

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 15 (As a user, I want to search for questions or answers)

Open Issues:

Acceptance tests: To test the search function, we need to first populate the question list and answer list with questions and answers. Then we use a simple search function where each element in the list is checked against the item we want. System assert will return true (test passing) if the item is indeed in the list and found. Similarly, answers are found in the same way, where the same search function is applied to a master answer list. Exception 3 is tested by running the search to find a string that is not in the built lists. System will check if the result is null (meaning not found).

14.

Use Case Name: QuestionsAsked

Participating Actors: Author

Goal: Device will remember the questions author asked

Trigger: Author submits a question

Preconditions: Author knows how to submit a question

Postconditions: On success, system saves the questions asked by author on to the device author is currently using

Basic Flow:

1. Author submits a question.
2. System save the question onto the device the author is currently using.
3. Author can view their questions asked under main menu AskedQuestions

Exceptions:**Qualities:****Constraints:****Includes:****Extends:****Related Artifacts:****Notes:**

User Story 16 (As an author, I want my device to remember which questions I asked)

Open Issues:

Acceptance tests: To test this, a question is created along with a picture. Then a new

question list is created and the question and picture is added to a question list. Lastly we check if the question list is empty or not to ensure the question is added successfully to the questionlist .

15.

Use Case Name: ReadingList

Participating Actors: User

Goal: Indicated questions and its answers will be locally cached and accessed without internet

Trigger: User chooses the AddToReadingList option

Preconditions: questions must exist and user device must have enough memory

Postconditions: On success, system saves the questions selected by user to the device and it can be accessed offline

Basic Flow:

1. User selects a question they want to add to reading list
2. System save the question and its answers onto the device the user is currently using.
3. User can view their reading list under main menu ReadingList.

Exceptions:

User Stories: 17

Additional

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 17 (As a user, I want questions and answers that I read or questions and answers that I've indicated I want to read, to be locally cached so I can read them when I am not on the internet.)

Open Issues:

Acceptance tests: To test User story 17, we first make a question and author and then add the question to an arraylist called "Reading list", then we check if there is anything in the arraylist to make sure we added the question to "Reading List".

16.

Use Case Name: Favorites

Participating Actors: User

Goal: Save questions as favorites and these questions and their replies can be accessed regardless of network activity

Trigger: User chooses the addFavorite option

Preconditions: At least one question must exist and user device must have enough memory

Postconditions: On success, system saves the questions and replies selected by user to the device and it can be accessed regardless of network activity

Basic Flow:

1. User selects a question they want to add to their favorite
2. System save the question and its answers onto the device the user is currently using.
3. User can view their favorite list under main menu.

Exceptions:

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 18 (As a user, I want to explicitly save some questions as favorites.)

User Story 19 (As a user, my favorites and their replies should always be available to me regardless of network connectivity.)

Open Issues:

Acceptance tests: To test user stories 18 and 19, we first make a question and author and then add the question to an arraylist called "Favorites", then we check if there is anything in the arraylist to make sure we added the question to "Favorites".

17.

Use Case Name: PushAnswers

Participating Actors: Author

Goal: Push replies, questions and answer wrote by the author online once the author gets internet connectivity

Trigger: Once there is network connection

Preconditions: author has replies, questions and answers written without internet connection

Postconditions: On success, system saves the questions and replies selected by user to the device and it can be accessed regardless of network activity

Basic Flow:

1. Author writes replies, answer, and/or questions without internet connection and press send
2. System displays “No network connection” message
3. Author finds network connection
4. System pushes replies, questions and answers online

Exceptions:

User Stories: 20, 21

Additional

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 20 (As an author, I want to author replies, questions and answers offline)

User Story 21 (As an author, I want to push my replies, questions and answers online once I get connectivity)

Open Issues:

Acceptance tests: To test user story 20, a question, answer, and reply is initiated and added to the question and answer list and the questions and answers are linked. Since this is a mock test a Boolean value (online) is first set to false indicating no internet connection. The system checks it and return “no network connection” , then the Boolean value is set to true indicating the user connecting to network and system will “push stuff online”.

18.

Use Case Name: SetUserName

Participating Actors: Author

Goal: Set a username that will be displayed when author post a question, reply or answer

Trigger: Author navigates to the setUserName page

Preconditions:

Postconditions: On success, system will set the author's username as the one typed in

Basic Flow:

1. Author types in his/her username
2. System displays his/her user name and wait for confirmation
3. Author confirms his/her username
4. System set the username and will display it after every post made by this user

Exceptions:

1. No username is typed in or only whitespace is typed in
 - 1.1 System set the Author as "Anonymous"

User Stories: 23

Additional

Qualities:

Constraints:

Includes:

Extends:

Related Artifacts:

Notes:

User Story 23 (As an author, I set my username)

Open Issues:

Acceptance tests: To test this, a method and test string 'Paul' are used to set the username of an author. To test exception 1 (empty string), an empty string is used and since it is empty, assert statement will display "no author specified, setting author to anonymous"

After setting the author, we check if the author set is 'Paul', we do not explicitly check if the author is set to anonymous because it is the same as checking if it is set to 'Paul'. Then the same test is repeated again but this time instead of setting the author to a question it is checking if the author is properly set to an answer.