# Test Document

## for

# MindHive

**Version 1.0**

**Prepared by**

**Group : 2** **Group Name:** MindHive

| | | |
|---|---|---|
| Aakash Om Trivedi | 200003 | aakashtri13@gmail.com |
| Adi Pratap Singh | 200036 | apsinghnsa@gmail.com |
| Bhavya Garg | 200270 | gargs.jpr@gmail.com |
| Gavish Garg | 200385 | gavish94170garg@gmail.com |
| Harsh Trivedi | 200422 | harsh.trivedi.mikki@gmail.com |
| L Gokulnath | 200542 | lgokulnath02@gmail.com |
| Parth Maniar | 200671 | pmaniar1906@gmail.com |
| Rishav Bikarwar | 200792 | ris27hav@gmail.com |
| Sahil Bansal | 200836 | sbansal.sb2002@gmail.com |
| Ujwal Jyot Panda | 201060 | ujwal12366@gmail.com |

# Contents

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# 1 Introduction

## Test Strategy

We have automated the unit testing and most parts of integration testing using Django's **test** framework. Django provides a built-in test library which inherits from the Python unittest library. Using the functionalities provided in this library we can do both unit testing and integration testing. The procedure for testing in Django is that we write some python files which contain the tests. Every time we make some changes in the code, the tests run automatically and inform us in case of any error. It is a bit similar to the Google Test framework which was discussed in class by the Professor.

The system testing was done manually, we tried to check all the functional and non-functional requirements of the SRS while doing the system testing, and also tried to observe the problems an end-user might face while using our software.

## When was the testing conducted?

The majority of the testing was done after the implementation was completed. However, even during the implementation phase, we frequently did some manual testing of our code. The automated tests were written after the implementation phase.

## The Testers

Since almost all of the team members were involved in the development process, we couldn't do independent testing. However, we ensured that the person who *wrote a particular functionality has not tested the same functionality*. While distributing the work for testing, we kept this in mind.

## Coverage Criteria

We have used functional coverage.

## Tools used for testing

As mentioned above, we have used the TestCase module from the django.test library to do unit testing.
For integration testing, we used the Selenium web framework.

Since the system testing involves lots of dynamic components, we did it manually.

# 2 Unit Testing

We created a object of class questions and then we have given it the attribute and then the test was run to check the given attributes

The fields/function names in square brackets were covered in the unit testing.

### 1. Question/Answer Forms

*[Question Title, Question* `Text,Question Author, Answer Text, Answer Author, Reporting Text`*]*

**Unit Details:** This test checks the functionality of forms used for taking input from the user. The inputs are provided to the form and are then saved. The values ate then cross verified from the saved form against the provided input.
**Test Owner**: *Adi Pratap Singh*
**Test Date**: *03/21/2021-03/21/2021*
**Test Results**: All test passed except report
**Structural Coverage:** Forms used for taking question/answer as input were tested for the correct data. Validity of form as a whole was tested by saving it into a temporary database. The temporary database was then cleared once the testing was completed.
**Additional Comments**: Currently the test for report was not automated so we will have to check that

### 2. Notification Model

*[QuestionModel, text_max_length, text_content, timestamp_content, receivers_content,target_questions]*

**Unit Details:** Model related constraints were verified in this unit, A custom object is defined and then it is fed to the model. The value associated with the model is then cross verified against the original object.
**Test Owner**: *Ujwal Jyot Panda*
**Test Date**: *03/22/2021-03/22/2021*
**Test Results**: All test passed.
**Structural Coverage:** Models used in the notifications class were verified in this unit. Data types, their respective values and maximum lengths were checked in this test
**Additional Comments**: *[Enter any summary comments]*

### 3. Question Model

*[Title_max_length,* `question_name, content_richfield, tag_check, likedBy_check, dislikedBy_check, viewedby_check, check_author,`

`check_anonymous, get_author_name, get_absolute_url, get_votes`*]*

**Unit Details***: Model related constraints were verified in this unit, A custom object is defined and then it is fed to the model. The value associated with the model is then crossverified against the original object.*
**Test Owner**: *Rishav Bikarwar*
**Test Date**: *03/22/2021-03/22/2021*
**Test Results**: All test passed.
**Structural Coverage:** Model used in the question class was tested in this module. Data types and their respective values were cross verified. At some places maximum length of valid data was also checked.
**Additional Comments**: *[Enter any summary comments]*

### *4. Tag Model*

*[setUp, name_label, name_max_length]*

**Unit Details***: For each question which is present in our database, a tag is associated with it. They act as an identifier for the question to other users. In this module we test the value associated with the tag and the maximum permitted size of value in a tag.*
**Test Owner**: *L Gokulnath*
**Test Date**: *03/24/2021-03/24/2021*
**Test Results**: All test passed.
**Structural Coverage:** *Model used in the Tags was covered in this module.*
**Additional Comments**: *[Enter any summary comments]*

### *5. Comment Model*

*[*`Text_content, author_username, check_annonymous, parentObjTypeQ, parent_ObjTypeA, parent_ObjQ, parent_ObjA, author_name`*]*

**Unit Details***:* Model related constraints were verified in this unit, A questions, answers and comment objects are defined and then are fed to the model. The value associated with the model is then crossverified against the original object.
**Test Owner**: *Parth Maniar*
**Test Date**: *03/24/2021-03/25/2021*
**Test Results**: All test passed.
**Structural Coverage:** Model used in the comment class was tested in this module. Values were cross verified with original object. At some places parent Object type and content of valid data was also checked.

### *6. Answer Model*

*[Content_richfield, tag_check, likedBy_check, dislikedBy_check, check_author, check_anonymous, get_author_name, vote_count]*

**Unit Details:** Model related constraints were verified in this unit, A custom object is defined and then it is fed to the model. The value associated with the model is then cross verified against the original object.

**Test Owner**: *Parth Maniar*

**Test Date**: *03/24/2021-03/25/2021*

**Test Results**: All test passed.

**Structural Coverage:** Model used in the Answer class was tested in this module. A custom object is defined and then it is fed to the model.

**Additional Comments**: *[Enter any summary comments]*

### *7. User Form*

*[UserCreateForm, UpdateUserInfo]*

**Unit Details:** In this unit, we have tested the form used for taking the input while user registration as well as Updation of data

**Test Owner**: *Sahil Bansal*

**Test Date**: *03/24/2021-03/25/2021*

**Test Results**: All test passed.

**Structural Coverage:** Form functionality related to user creation and data updation was tested in this module.

**Additional Comments**: *[Enter any summary comments]*

### *8. User Model*

*[email_label, email_max_length, username_max_length, roll_no_max_length, name_max_length]*

**Unit Details:** In this unit, we have tested the models used in user class. A user object is defined and then it is fed to the model. The value associated with the model is then crossverified against the original object.

**Test Owner**: *Ujwal Jyot Panda*

**Test Date**: *03/24/2021-03/25/2021*

**Test Results**: All test passed.

**Structural Coverage:** Model used in the user class was tested in this module. Values were cross verified with the original object. At some places, max_length of various models are checked.

### 9. Question View

*[view_question]*

**Unit Details:** In this unit test, we have tested the view that displays the question details and provides buttons for various functionalities related to question

**Test Owner**: *Rishav Bikarwar*

**Test Date**: *03/25/2021-03/25/2021*

**Test Results**: All test passed.

**Structural Coverage:** View functionality related to URL validation, correct template usage, redirection if not logged in and data validation was tested in this module.

### 10. Question Edit View

*[edit_question]*

**Unit Details:** In this unit test, we have tested the view that provides functionality for editing an already posted question.

**Test Owner**: *Rishav Bikarwar*

**Test Date**: *03/25/2021-03/25/2021*

**Test Results**: All test passed.

**Structural Coverage:** View functionality related to URL validation, redirection if not logged in, author validation and data validation was tested in this module.

### 11. Vote View

*[vote]*

**Unit Details:** In this unit, we have tested the view that provides the functionality for upvoting or downvoting questions and comments

**Test Owner**: *Rishav Bikarwar*

**Test Date**: *03/25/2021-03/25/2021*

**Test Results**: All test passed.

**Structural Coverage:** View functionality related to URL validation, redirection if not logged in and vote functionality validation was tested in this module.

### 12. Follow Bookmark View

*[follow_bookmark]*

**Unit Details:** In this unit, we have tested the view that provides the functionality for following or bookmarking a question

**Test Owner**: *Rishav Bikarwar*

**Test Date**: *03/25/2021-03/25/2021*

**Test Results**: All test passed.
**Structural Coverage:** View functionality related to URL validation, redirection if not logged in and bookmark functionality validation was tested in this module.

## *13. Add Comment View*

*[view_comment]*

**Unit Details:** In this unit, we have tested the view that provides the functionality for adding a comment on any question or comment from the question details view
**Test Owner**: *Rishav Bikarwar*
**Test Date**: *03/26/2021-03/26/2021*
**Test Results**: All test passed.
**Structural Coverage:** View functionality related to URL validation, correct template usage, redirection if not logged in and data validation was tested in this module.

# 3 Integration Testing

(Note: Logout before running any of the below test)

## 1. Login functionality

**Module Details:** This module contains all classes, templates and functions necessary for user authentication while logging in.

**Test Owner**: Harsh Trivedi

**Test Date**: 26/03/2022 - 26/03/2022

**Test Results** All tests passed.

**Additional Comments**: : Using the selenium framework, code was written to open the login page of our software and also auto-fill the credentials (email and password). Both valid and invalid credentials were tried and correct results were obtained.

**Command:** python3 test_wrongusernamepasswordforgotpassword.py

## 2. Sign up

**Module Details:** This module contains all classes, templates and functions necessary for user authentication while signing up

**Test Owner**:  Parth Maniar

**Test Date**: 26/03/2022 - 27/03/2022

**Test Results**: Passed

**Additional Comments**: Implementation was similar to login. Tried adding existing users, new users, and checked all the password validation tools also.

**Command:** python3 test_signuptests.py
: python3 test_newuserfromsigninpage.py

## 3. Ask Question

**Module Details:** It contains the form for asking questions, submitting the question, and view the asked question

**Test Owner**: Harsh Trivedi

**Test Date**: 26/03/2022 - 27/03/2022

**Test Results**: No error showing on leaving the question text field blank (but still the question won't be submitted). Upload image from local system option isn't visible. In edit question, the time doesn't get updated when one edits the question.And we have also tested the anonymous by login from other id.

**Additional Comments**: Anonymous question feature is also working fine. Have corrected the time feature now.

**Command:** python3 test_questionnormal.py

: python3 test_askquestionanonymous.py (to check ask questions anonymous)

: python3 test_askquestionerror.py (to simulate error if given wrong input)

### *4.* Add Answer to Question (error in automated testing)

**Module Details:** It contains the form for answering the previously asked questions.

**Test Owner**: L Gokulnath

**Test Date**: 28/03/2022 - 28/03/2022

**Test Results:** Time is showing up incorrectly, time of asking question is showing up in the answer also, not the time of answering the question. Rest everything works fine. Corrected the time issue on 1st April. There is some iframe error in this so we are working on it automated test.

**Additional Comments**: Anonymous answer feature is also working fine. Corrected the time issue on 1st April

**Command:** python3 test_addanswernormalwithaddquestion.py

### *5. Add Comment*

**Module Details:** Simple text box to add comments to questions/answers

**Test Owner**: L Gokulnath

**Test Date**: 28/03/2022 - 28/03/2022

**Test Results**: Same time issue as in Add answer to question

**Additional Comments**: Corrected the time issue on 1st April

**Command:** python3 test_addingcomment.py

### *6.* Edit Profile

**Module Details:** Users have the option to edit their profile picture and username in this module.

**Test Owner**: L Gokulnath

**Test Date**: 28/03/2022 - 28/03/2022

**Test Results**: No error showing up on editing the username to an already existing username. Rest everything works fine.

**Additional Comments**: NA

### *7.* Like, Dislike questions/answers

**Module Details:** Users can like/dislike questions, answers

**Test Owner**: Adi Pratap Singh

**Test Date**: 29/03/2022 - 29/03/2022

**Test Results**: Passed

**Additional Comments**: Tried liking, disliking, liking and then disliking both questions and answers. Logged in as another user to check if the changes reflect correctly for that user also.

**Command:** python3 test_likedisliketest.py

### *8.* Notifications and follow question functionality

**Module Details:** Users get notifications when the questions asked by them have received answers

**Test Owner**: Adi Pratap Singh

**Test Date**: 29/03/2022 - 29/03/2022

**Test Results**: Works fine

**Additional Comments**: Same notification shows up repeatedly on login, need to click on each notification to make it disappear. Maybe there can be a clear notifications option like in Mobile phones.

**Command:** python3 test_follownotificationtest.py

### *9.* Add Favourite Tags

**Module Details:** Users can add/edit their favourite tags, and

**Test Owner**: Adi Pratap Singh

**Test Date**: 29/03/2022 - 29/03/2022

**Test Results**: Passed

**Additional Comments**: Tried liking, disliking, liking and then disliking both questions and answers. Logged in as another user to check if the changes reflect correctly for that user also.

### *10.* Add Report Test

**Module Details:** We check whether users can report question and answers

**Test Owner**: Adi Pratap Singh

**Test Date**: 29/03/2022 - 29/03/2022

**Test Results**: Passed

**Additional Comments**: We first created a question then reported it but we have to manually check on the admin page that we get the report as we don’t want our test to have admin details.
**Command:** python3 test_reportquestionchecks.py

### *11.* Bookmark test

**Module Details:** Added the question to bookmark and check both functionality.
**Test Owner**: Adi Pratap Singh
**Test Date**: 30/03/2022 - 30/03/2022
**Test Results**: Passed
**Additional Comments**: First created a dummy question then added it to the bookmark question then checked the list.
**Command:** python3 test_addbookmarkquestion.py

# 4 System Testing

Here, we focus on the software more from an end-user point of view.

1. **Requirement:** Site launches properly, all images and text are rendered correctly
**Test Owner**: Ujwal Jyot Panda
**Test Date**:29/3/2022 - 29/3/2022
**Test Results**: Site launched properly with all text and images rendered as expected
**Additional Comments**: The site was launched in localhost from our system. We tried launching the site on all the three popular OSs (Windows, Linux and macOS) and all the popular browsers.

2. **Requirement:** Users can register/login to the site
**Test Owner**: Ujwal Jyot Panda
**Test Date**: 29/03/2022 - 30/3/2022
**Test Results**: At first the login functionality was tested using the credentials of the users who had already been registered in the website. Registered users can login successfully if they give the correct credentials. The forgot password functionality was also tested and we tried logging in using the new password. It too worked as expected. Finally, we registered a new user using the sign up functionality, after which the website redirected to the login page. We tried logging that user in. This also worked as expected.
**Additional Comments**: We also tried signing up a user with credentials of an already existing user. The site showed a warning and didn't allow the user to register. The forgot password email does take some time to show up on our inboxes. In the future, it's better to set up our own SMTP server.

3. **Requirement:** Maintain User profile
**Test Owner**: Gavish Garg
**Test Date**: 30/03/2022 - 30/3/2022
**Test Results**: Registering a new user has already been tested in the above test. Here we basically checked the view profile and edit profile functionalities. So, we logged in to the site and viewed the profile of a user. The results were as expected, the correct credentials and the set profile picture showed up. Then, edit profile functionality was tested. A new profile pic was added, and then we viewed the profile of the user. The changes reflect over there.
**Additional Comments**: NA

4. **Requirement:** Provide different fields of interests (tags) and Question categorization

**Test Owner**: Bhavya Garg
**Test Date**: 30/03/2022 - 30/3/2022
**Test Results**: Tested the add/edit favourite tags functionality. Some tags were randomly selected as favourite and some already selected tags were deselected. The changes reflect correctly on the homepage, under the favourite tags box. Also, on clicking the tags in the popular/favourite tags, one should be redirected to a page which contains all the questions which are associated with those tags. This too worked as expected.
**Additional Comments**: Currently, a normal user can't create tags on his/her own. The tags are predefined by the admin. One of the future goals can be to allow users to add relevant tags on their own.

5. **Requirement:** A home page to show previously asked questions

**Test Owner**: L Gokulnath
**Test Date**: 31/03/2022 - 31/3/2022
**Test Results**:
Tests conducted:

1. On logging in, the user should be redirected to his/her home page which shows all the questions which are associated with the favourite tags of the user.
2. On editing the favourite tags, the home page should update automatically to reflect the additional questions/remove some questions as per the requirement.

Test results:

1. Test passed successfully.
2. Test passed successfully.

**Additional Comments**: NA

6. **Requirement:** Users can ask questions

**Test Owner**: L Gokulnath
**Test Date**: 30/03/2022 - 31/3/2022
**Test Results**:
Tests conducted:

1. An authenticated user can ask questions using the ask questions button on the home page, and it should reflect on the asked questions link of the user, and in the questions list of all the other users.
2. User can select appropriate tags associated with the question to help in better classification. After submitting the question, it should be visible under the tags header of whichever tags were selected for this question.

3. Like/dislike the question, and see if it reflects correctly when another user has logged in.

Results:

1. Test passed successfully
2. Test passed successfully
3. Test passed successfully

**Additional Comments**: Users can also add rich text, images, hyperlinks, etc. in their questions. These things were also tested.

**7. Requirement:** Search question facility

**Test Owner**: Rishav

**Test Date**: 1/04/2022 - 1/4/2022

**Test Results**:

Tests conducted:

1. On entering the appropriate keyword (in the text/title of the question), the search results should display the relevant questions with matching keywords.

Result:

1. Test passed successfully

**Additional Comments**: NA

**8. Requirement:** Answer questions previously asked

**Test Owner**: Rishav

**Test Date**: 1/04/2022 - 1/4/2022

**Test Results**:

Tests conducted:

1. Add answers to previously asked questions, like/dislike the answers.
2. See if these changes reflect correctly on the view question page of a particular question.

Test results: All results are as expected

**Additional Comments**: NA

**9. Requirement:** Add comments on questions and answers

**Test Owner**: Rishav

**Test Date**: 1/04/2022 - 1/4/2022

**Test Results**:

Tests conducted:

1. Add comments to previously asked questions and answers.

2. Login as another user and see if the comments reflect them

Test results: All results are as expected

**Additional Comments**: NA

**10. Requirement:** Provide simple facility for reporting a particular question

**Test Owner**: Adi

**Test Date**: 2/04/2022 - 2/4/2022

**Test Results**:

Tests conducted:

1. Reported an already existing question, created a new question and reported it.
2. Accessed the report text from the admin interface, and removed the reported questions

Test results: All results are as expected.

**Additional Comments**: NA

# 5 Conclusion

**How Effective and exhaustive was the testing?**
The unit testing and most parts of integration testing are automated, so they are highly effective. We have tried to maximize the exhaustiveness of our testing process.The unit testing has been done quite extensively, covering almost all the functions. Integration testing was done for most of the connected modules. System testing was done for all the requirements of the SRS which have been implemented by us.

**Which components have not been tested adequately?**
For the system testing we can’t test it automatically and also the add answer in the integration testing is not correct due to some iframe error. Rest all is tested correctly.

**What difficulties have you faced during testing?**
Automating integration testing and system testing was a challenge. For integration testing, we used the Selenium framework, but since system testing involves many testing interrelated components, we decided to go for manual system testing.

**How could the testing process be improved?**

- The obvious way of improving the testing process would be to have a separate testing team (no developer should be involved in the testing process).
- The developer might subconsciously assume a few crucial things which may be not so obvious to an end-user or someone who was not involved in the development of the software.
- Using structural coverage would have led to more exhaustive testing, but due to time constraints, we settled for functional testing.

# Appendix A - Group Log

| MEET DATE | TOPIC DISCUSSED | DURATION |
|---|---|---|
| 21 March 2022 | We discussed how to do the testing process for the website and the work distribution | 90 minutes |
| 22 March 2022 | Started Unit testing | 120 minutes |
| 25 March 2022 | Started Integration Testing | 120 minutes |
| 29 March 2022 | Started System Testing | 60 minutes |
| 30 March 2022 | System Testing Continued and finalized User Manual | 120 minutes |
| 3 April 2022 | Final Document Compilation | 60 minutes |