

## Team Readly Members

1. Aviral Mansingka
2. Nihanshu Purohit
3. Sidhant Chadda
4. Sidhant Chitkara
5. **Sayali Aniket Kate (Project Co-ordinator)**

# 1 Facebook Login

## 1.1 Functional Testing

1. **Facebook 001**, Facebook Login Button check. Severity, important.
2. Click on the 'Facebook Login' button.
3. Redirects user to the page asking user's login credential.

## 1.2 Equivalence Class Testing

1. **Facebook 002**, Verifying the combination of Facebook credentials. Severity, workaround.
2. Click on 'login' button with a correct combination of Facebook login details.
3. Successful login and redirects the user to the home page.

## 1.3 Boundary Value Testing

1. **Facebook 003**, no username. Severity, workaround.
  2. Click on 'login' button with no username field.
  3. An error saying username is required.
- 
1. **Facebook 004**, no password or incorrect password. Severity, workaround.
  2. Click on 'login' button with no password or incorrect password field.
  3. An error saying correct password is required.

## 2 Google login

### 2.1 Functional Testing

1. **Google 001**, Google Login Button. Severity, important.
2. Click on 'Google Login' button
3. Redirects user to the page asking user's login credential.

### 2.2 Equivalence Class Testing

1. **Google 002**, Verifying the combination of google credentials. Severity workaround.
2. Click on 'login' button with a correct combination of google login credentials
3. Successful login and redirects the user to the home page.

### 2.3 Boundary Value Testing

1. **Google 003**, no password or incorrect password, Severity workaround.
2. Click on 'login' button with no password or incorrect password field.
3. An error saying correct password is required.

## 3 Trending News

### 3.1 Functional Testing

1. **Trending 001**, Click on trending button. Severity, important.
2. Click on the trending news tab.
3. Display all the current trending news articles.

1. **Trending 002**, Refresh, Severity, important.
2. Refresh trending news tab.
3. Server sends updated content to page.

### 3.2 Equivalence Class Testing

1. **Trending 003**. Equally weighted preferences, Severity workaround
2. Click on trending tab when preferences equally weigh 2 categories.
3. Show article that more views.

### 3.3 Boundary Value Testing

1. **Trending 004**. No users. Severity, workaround.
2. Click on trending tab when no users have used the system.
3. Text should appear on the screen stating that no news is currently trending.

## 4 Upvote

### 4.1 Functional Testing

1. **Upvote 001.** Upvote news article button. Severity, critical.
  2. Click upvote button in news article
  3. Upvote button should change color indicating that you have clicked on the button.
- 
1. **Upvote 002.** Undo upvote news article button. Severity critical.
  2. Click upvote button in news article after already having pressed it.
  3. Upvote button should change color indicating that you have clicked on the button.

### 4.2 Equivalence Class Testing

1. **Upvote 003.** Upvote count increase. Severity, important.
  2. Click upvote button in news article.
  3. Upvote count should increase after user interaction.
- 
1. **Upvote 004.** Undo upvote count decrease. Severity, important.
  2. Click upvote button in news article after already having pressed it.
  3. Upvote count should increase after user interaction.

### 4.3 Boundary Value Testing

1. **Upvote 005.** Upvote count non-zero. Severity workaround.
2. Click upvote when an article has no upvotes.
3. Count should go up from zero, and undoing should put it back to 0.

## 5 Downvote

### 5.1 Functional Testing

1. **Downvote 001.** Downvote news article button. Severity, important.
  2. Click Downvote button in news article
  3. Downvote button should change color indicating that you have clicked on the button.
- 
1. **Downvote 002.** Undo Downvote news article button. Severity critical.
  2. Click Downvote button in news article after already having pressed it.
  3. Downvote button should change color indicating that you have clicked on the button.

### 5.2 Equivalence Class Testing

1. **Downvote 003.** Downvote count increase. Severity important.
  2. Click Downvote button in news article.
  3. Downvote count should increase after user interaction.
- 
1. **Downvote 004.** Undo Downvote count decrease. Severity important.
  2. Click Downvote button in news article after already having pressed it.
  3. Downvote count should increase after user interaction.

### 5.3 Boundary Value Testing

1. **Downvote 005.** Downvote count non-zero. Severity workaround
2. Click Downvote when an article has no Downvotes.
3. Count should go up from zero, and undoing should put it back to 0.

## 6 Initially Recommended Articles

### 6.1 Functional Testing

1. **Initial Recommend 001.** Show recommended content. Severity, critical.
2. Open main webpage and scroll through the feed of articles.
3. News articles should be recommended based on your genre preferences.

### 6.2 Equivalence Class Testing

1. **Initial Recommend 002.** Show strongly recommended content. Severity, workaround
2. Open main webpage and scroll through the feed of articles.
3. Even if user only wants exactly one genre, we will occasionally show him other articles.

### 6.3 Boundary Value Testing

1. **Initial Recommend 003.** No preferences in system. Severity, workaround
2. Open main webpage and scroll through the feed of articles.
3. Show users trending articles since no information is present yet.

## 7 Estimated Reading Time

### 7.1 Functional Testing

1. **Reading 001.** Estimate reading time of page. Severity, important.
2. Click time icon on page of article.
3. Show user estimated time to read article.

### 7.2 Equivalence Class Testing

1. **Reading 002.** Large or small text with no image. Severity, workaround
2. Click time icon on page of article.
3. Make estimates using a simple formula.

1. **Reading 003.** Only image. Severity, workaround
2. Click time icon on page of article.
3. Make estimates using a count of images.

### 7.3 Boundary Value Testing

1. **Reading 004.** Near blank article. Severity, workaround
2. Click time icon on page of article.
3. Show user time estimate not available.



## 8 Recommend Articles based on usage

### 8.1 Functional Testing

1. **Usage Recommendation 001.** Track genre preference. Severity, critical.
2. User opens main page and scrolls through feed.
3. System recommends articles based on previously genre browsing.

### 8.2 Equivalence Class Testing

1. **Usage Recommendation 002.** User abstains from future voting. Severity, workaround
2. User avoid voting on relevant articles.
3. Slowly reduce all preference values to defaults.

### 8.3 Boundary Value Testing

1. **Usage Recommendation 003.** Vote relevant on every article. Severity, workaround
2. User votes relevant on every article.
3. Continue to recommend articles based on their previous preferences.

## 9 Recommend Articles based on strength

### 9.1 Functional Testing

1. **Strength Recommendation 001.** Filter Strength Checkbox. Severity, important.
2. Click 'filter by strength' checkbox.
3. Change color of checkbox to indicate that selection has been made.

### 9.2 Equivalence Class Testing

1. **Strength Recommendation 002.** Check Strength Recommendation. Severity, critical.
2. Open main page and scroll feed.
3. Show blend of user recommended stories and trending articles.

### 9.3 Boundary Value Testing

1. **Strength Recommendation 003.** Empty Strength Recommendation. Severity, workaround
2. Open main page and scroll feed.
3. If no user preferences have been stored, simply show normal trending articles.

## 10 Recommend Articles based on length

### 10.1 Functional Testing

1. **Length Recommendation 001.** Checking the functionality of the clock icon. Severity, important.
2. Click the 'clock' icon.
3. Reading time estimate displays next to the clock icon.

### 10.2 Equivalence Class Testing

1. **Length Recommendation 002.** Checking the length of an article. Severity, workaround
2. Click the 'clock' icon.
3. System gives the user reading time estimates for each article.

### 10.3 Boundary Value Testing

1. **Length Recommendation 003.** Ensure the clock icon only displays next to article titles. Severity, important.
  2. Look at the article panel heading.
  3. The clock should be next to the article heading.
- 
1. **Length Recommendation 004.** Ensure the clock icon only displays reading time estimates when moused over. Severity, important
  2. Mouse over the clock icon.
  3. The reading time estimate should be displayed.

## 11 Multiple sources

### 11.1 Functional Testing

1. **Multiple Sources 001**, Check for multiple news sources for a particular content. Severity, critical.
2. Click on 'other sources' button.
3. System should display multiple links of the sources.

1. **Multiple Sources 002**, Ensure the redirection of the page to the article's original page. Severity, critical.
2. Click on the article heading.
3. System should redirect you to the article's original page.

### 11.2 Equivalence Class Testing

1. **Multiple Sources 003**, Ensure the correct redirection of the page to the selected source. Severity, critical.
2. Click on the source link.
3. System should redirect you to the article's page by the corresponding source.

### 11.3 Boundary Value Testing

1. **Sources 004**, Ensure the redirection only happens when the user clicks on the article's heading. Severity, critical.
2. Click anywhere or any icon except the heading.
3. System shouldn't redirect you to the article's original page.

## 12 Relevance Rating

### 12.1 Functional Testing

1. **Relevance Rating 001**, Ensure the user is able to provide a relevance rating. Severity, critical.
2. Click on the "rating" button.
3. System should let the user rate the article's relevance.

1. **Relevance Rating 002**, Notification on successfully submitting the feedback. Severity, workaround
2. Click on the "submit" button.
3. System should notify the user with a success response.

### 12.2 Equivalence Class Testing

1. **Relevance Rating 003**, Check if the user is able to rate relevant or not. Severity, critical.
2. Select any number of stars between 0-5.
3. System should store the correct rating in the database.

### 12.3 Boundary Value Testing

1. **Relevance Rating 004**, Error notification. Severity, workaround
2. Click on the "submit" button without rating it.
3. System should notify the user with an error response.

## 13 Facebook Sharing

### 13.1 Functional Testing

1. **Facebook Sharing 001**, Ensure the user is able to click on the Facebook share icon. Severity, important.
2. Click on the "facebook share" icon.
3. System should allow the user continue with the facebook sharing process.

1. **Facebook Sharing 002**, Store facebook credential. Severity, workaround
2. Login with a facebook account and Click on the "Facebook share" icon.
3. The system should redirect the user to the facebook sharing process.

### 13.2 Equivalence Class Testing

1. **Facebook Sharing 003**, Ensure the user is able to share on facebook. Severity, workaround
2. Click on the "facebook share" icon.
3. The article should be on your Facebook wall.

### 13.3 Boundary Value Testing

1. **Facebook Sharing 004**, Redirect facebook on failed SignIn. Severity, workaround
2. Login with a Google account and Click on the "Facebook share" icon.
3. The system should redirect the user to the facebook login page.

## 14 Comments

### 14.1 Functional Testing

1. **Comments 001**, Ensure the user is able to click on the comment button. Severity, important.
2. Click on the "Comment" button.
3. System should display a text box linked to the article.

1. **Comments 002**, Ensure the user is able to type in the comment text box. Severity, important.
2. Start typing in the text box.
3. System should allow you type in the text box.

### 14.2 Equivalence Class Testing

1. **Comments 003**, Ensure the user is able to successfully post the comment on an article. Severity, important.
2. Type up a comment and click on the "done" button.
3. System should post the comment linked to the selected article.

1. **Comments 004**, Ensure the user is able to delete the comment on an article. Severity, workaround
2. Click on the "delete" icon.
3. System should delete the selected comment.

### 14.3 Boundary Value Testing

1. **Comments 005**, Ensure the user does not submit a blank entry field. Severity, important.
2. Don't type and try to click "done".
3. Submit button shouldn't be click-able.

## 15 Blacklist

### 15.1 Functional Testing

1. **Blacklist 001**, Ensure the user is able to click on the block button. Severity, important.
  2. Click on the "block" button.
  3. A pop-up box appears to display a confirmation message.
- 
1. **Blacklist 002**, Ensure the confirmation button is click-able. Severity, important.
  2. Click on the "Confirm" button.
  3. The dialog box should disappear.

### 15.2 Equivalence Class Testing

1. **Blacklist 003**, Ensure the correct source is blocked. Severity, critical.
2. Click on the "block" button and then the confirmation button.
3. System shouldn't display any news from that particular source.

### 15.3 Boundary Value Testing

1. **Blacklist 004**, Ensuring the source is blocked after the user's confirmation. Severity, workaround
2. Click on the "block" button and then the "cancel" button.
3. System shouldn't block the source.



## 16 Subscribe

### 16.1 Functional Testing

1. **Subscribe 001**, User Subscribes to a particular topic, Severity important.
2. User clicks on subscribe on a news article
3. User is alerted when a news article on a similar topic arises.

### 16.2 Equivalence Class Testing

1. **Subscribe 002**, User subscribes to multiple topics, Severity important.
2. User subscribes to several different topics.
3. User is alerted whenever a article is is written on any of these topics.

### 16.3 Boundary Value Testing

1. **Subscribe 003**, User subscribes to inactive topic, Severity workaround
2. User Subscribes to a topic that is no longer being reported on by news stations.
3. User is not notified at all.

## 17 Favorite

### 17.1 Functional Testing

1. **Favorite 001**, Favorite Button, Severity, critical
2. Click on favorite button in a news article
3. Favorite button should visually change indicating that the button has been pressed

### 17.2 Equivalence Class Testing

1. **Favorite 002**, Favorite by applying collaborative filtering, Severity, important
2. User favorites an article.
3. User is recommended news articles that relate to the article they have favorited.

### 17.3 Boundary Value Testing

1. **Favorite 003**, Favorite Broken link, Severity, 3
2. User favorites an article that has been deleted
3. User is still recommend news articles that relate to the article they have favorited.

## 18 Click to Read

### 18.1 Functional Testing

1. **Click To Read 001**, User clicks button to read more about article. Severity, critical.
2. User clicks on button to read more about the specific article
3. User is redirected to to a page which contains more information about the specific article.

### 18.2 Equivalence Class Testing

1. **Click to Read002**, User clicks multiple times. Severity important.
2. User clicks on button to read more about an article multiple times.
3. User is presented with the same information every time.

### 18.3 Boundary Value Testing

1. **Click To Read 003**, User clicks broken link. Severity, important.
2. User clicks to view an article that has been deleted.
3. User is redirected to page stating that the article has been deleted.

## 19 Headlines

### 19.1 Functional Testing

1. **Headlines 001**, View Headlines. Severity, critical.
2. User looks at article in news feed
3. User can look at the headline of the article.

### 19.2 Equivalence Class Testing

1. **Headlines 002**, Search by headlines, Severity important
2. User searches for a headline
3. User is presented with multiple similar headlines.

### 19.3 Boundary Value Testing

1. **Headlines 003**, No Headline. Severity, important.
2. User views an article that does not have a headline.
3. User is given the appropriate information about the article without information about the headline.

## **20 Contact Developers**

### **20.1 Functional Testing**

1. **Contact Devs 001**, User submits content. Severity, workaround
2. User submits form to the developers, in response to the user's using the platform.
3. User receives confirmation that email has been sent to developers.

### **20.2 Equivalence Class Testing**

1. **Contact Devs 002**, Multiple emails, Severity workaround.
2. Multiple users submit form to developers.
3. Each user receives form acknowledging that they have submitted a form to the developer

### **20.3 Boundary Value Testing**

1. **Contact Devs 003**, User submits blank form. Severity, workaround.
2. User submits blank form to the developers.
3. The developers are not emailed anything.

## **21 Report**

### **21.1 Functional Testing**

1. **Report 001**, User clicks report button. Severity, critical.
2. User reports another user for posting inappropriate content.
3. Reported User is blocked from the users account

### **21.2 Equivalence Class Testing**

1. **Report 002**, Multiple reports, Severity important
2. Multiple users report the same user.
3. That user's account is temporarily banned.

### **21.3 Boundary Value Testing**

1. **Report 002**, User blocks deleted account. Severity, important.
2. User blocks another user who has deleted their account.
3. Blocked user's posting should be deleted.

## **22 Save**

### **22.1 Functional Testing**

1. **Save 001**, User saves article. Severity, critical.
2. User clicks on save button on an article. And views saved articles tab
3. Users can see their newly saved article in this list.

### **22.2 Equivalence Class Testing**

1. **Save 002**, User Saves multiple articles, Severity important.
2. User saves multiple articles
3. Users can see all their saved articles on saved articles tab.

### **22.3 Boundary Value Testing**

1. **Save 003**, User saves a deleted article. Severity, workaround.
2. User saves a deleted article to their saved list.
3. User is unable to view the article on their saved list tab.

## 23 Saved News Organized

### 23.1 Functional Testing

1. **Saved Organized 001**, User saves an article. Severity, important.
2. User clicks on save an article.
3. That article is placed into that users saved articles under a specific category

### 23.2 Equivalence Class Testing

1. **Saved Organized 002**, User saves two similar articles. Severity, workaround.
2. User saves two articles that relate in topics
3. Both articles are under the same category under user's saved news.

1. **Saved Organized 003**, User saves two different articles. Severity, important.
2. User saves two articles that do not relate
3. Articles are saved in the saved categories in two separate categories.

### 23.3 Boundary Value Testing

1. **Saved Organized 003**, User saves two somewhat different articles. Severity, workaround.
2. User Saves two articles that talk about similar things but differ in content.
3. The articles are saved under different categories in the saved news section.



## 24 Images and Text

### 24.1 Functional Testing

1. **Images and Text 001**, Category selection, Severity, important.
2. User clicks on a specific category in the news section.
3. User should be able to view text and images relating to that specific category.

### 24.2 Equivalence Class Testing

1. **Images and Text 002**, User clicks on category with no images, Severity workaround.
2. User clicks on a category that does not have any images relating to it yet.
3. User is not presented with any images when they click on that category.

### 24.3 Boundary Value Testing

1. **Images and Text003**, User clicks on blank category, Severity workaround.
2. User clicks on category that has no articles relating to it.
3. User is not presented with any images or text upon clicking on the category.

## 25 Settings

### 25.1 Functional Testing

1. **Settings 001**, Settings tab, Severity critical
2. User clicks on settings tab
3. User is presented with a list of their current settings/configurations.

1. **Settings 002**, Settings Save, Severity important
2. User adjusts settings inside settings tab
3. User's web page reflects these changed settings

### 25.2 Equivalence Class Testing

1. **Settings 003**, User Delete, Severity critical
2. User requests to delete their account inside settings tab.
3. User account is unable to access their account.

### 25.3 Boundary Value Testing

1. **Settings 004**, Delete Unused Account, Severity critical
2. User requests to delete their account even though no information regarding their user preferences are attached to their account.
3. User is still able to delete their account.