

# Group 29 (AKT)

## Test Plan

Alex Trudeau  
400030148

Kathryn Kodama  
400013582

Tommy Tran  
001150067

October 28, 2017  
Version 1.3

## Contents

<b>1</b>	<b>General Information</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.3	Acronyms, Abbreviations, and Symbols . . . . .	3
<b>2</b>	<b>Plan</b>	<b>4</b>
2.1	Software Description . . . . .	4
2.2	Test Team . . . . .	4
2.3	Tools Used for Testing . . . . .	4
2.4	Testing Schedule . . . . .	5
<b>3</b>	<b>System Testing</b>	<b>6</b>
3.1	Tests for Functional Requirements . . . . .	6
3.1.1	Account Management . . . . .	6
3.1.2	Content Management . . . . .	8
3.1.3	Navigation Methods . . . . .	13
3.2	Tests for Nonfunctional Requirements . . . . .	14
3.2.1	Portability . . . . .	14
3.2.2	Performance . . . . .	15
3.2.3	Security . . . . .	15
<b>4</b>	<b>Comparison to Existing Implementation</b>	<b>16</b>
<b>5</b>	<b>Unit Testing Plan</b>	<b>16</b>
5.1	Unit testing of internal functions . . . . .	16

## List of Tables

1	Revision History . . . . .	3
2	Table of Abbreviations . . . . .	3
3	Table of Definitions . . . . .	4
4	Testing Schedule (as outlined in the Gantt Chart) . . . . .	5

Table 1: **Revision History**

Version	Date	Notes
1.0	18/10/17	Created Document
1.1	24/10/17	Updated content and formatting
1.2	26/10/17	Updated content
1.3	27/10/17	Finalized content for rev.0

\*list of figures is not applicable to this document as of version 1.3

# 1 General Information

## 1.1 Purpose

The purpose of this test plan is to verify that the program works as intended and was implemented correctly with the utmost confidence.

## 1.2 Scope

This test plan provides the basis for testing the Reddit-Clone program's functionality and that it properly mimics the functionality of the original open source project Reddit. The program allows for users to post, discuss, and curate user created content.

The testing of Reddit-Clone means to cover Dynamic testing and dynamic testing through black box, white box, manual, and automated tests. This document is an outline of the testing methods including the testing tools to be utilized.

## 1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	Definition
AKT	Alex, Kathryn, Tommy; also the team name (see Definitions table)
PoC	Proof of concept, project as of October 16, 2017
DevPlan	Development Plan document
UI	User Interaction

Table 3: **Table of Definitions**

Term	Definition
Reddit	The original Reddit open source project to be recreated ( <a href="http://www.reddit.com">http://www.reddit.com</a> )
Reddit-Clone	The project being created.
AKT	The developer team for Reddit-Clone
Up Vote	A term for "liking" a post/comment on Reddit-Clone.
Down Vote	A term for "disliking" a post/comment on Reddit-Clone.
Subreddit	A page containing the posts of a certain topic
GUI	Graphic User Interface
Post	A submission by an authenticated user that contains text and/or media attachments
Comment	An authenticated user's text response to another user's post or comment

## 2 Plan

This section provides information on the Reddit-Clone project and the software testing plan. This plan is based off of the main functionality that exists as of the PoC as well as any additional functions outlined in the DevPlan.

### 2.1 Software Description

Reddit-Clone will allow authenticated users create posts, subreddits and comments as well as curate others' content through up-votes and down-votes. The program is implemented in JavaScript with Angular 4 and Ionic 2 framework.

### 2.2 Test Team

The testing team for Reddit-Clone are all members of the AKT Development team: Alex Trudeau, Kathryn Kodama, Tommy Tran. The lead tester will be Tommy Tran.

### 2.3 Tools Used for Testing

The dynamic testing will utilize the following tools. Mocha and Angular 4 testing packages will be used to test the JavaScript functionality of this project.

The Dynamic testing will be performed without additional tools. The HTML/Css testing will be tested through observation on Unix, Windows, iOS, and Android environments.

## 2.4 Testing Schedule

Refer to the [Gantt Chart](#) located in the project GitLab for the testing schema. Testing is divided into the following categories within the Gantt chart: Account Management, Content Management, Navigation Methods. Each testing category is assigned to a different member of the AKT development team and associated with a start and date time

Table 4: **Testing Schedule (as outlined in the Gantt Chart)**

Category	Group Member	Completion Date
Account Management	Alex Trudeau	7/11/2017
Content Management	Tommy Tran	9/11/2017
Navigation Methods	Kathryn Kodama	11/11/2017

## 3 System Testing

### 3.1 Tests for Functional Requirements

#### 3.1.1 Account Management

##### User account creation (valid inputs)

**Test ID:** FRT1-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Not authenticated, on home page

**Input:** Enter valid username, email and password

**Output:** Email sent to inputted email(user cannot log in yet).

**How the test will be performed:** Tester will sign up and after successful sign up, check that an email was received. Database will also be verified for successful user creation.

##### User account creation (invalid inputs)

**Test ID:** FRT1-2

**Type:** Functional, Dynamic, Manual

**Initial State:** Not authenticated, on home page

**Input:** Enter invalid or already registered username/email

**Output:** Error message shown to user

**How the test will be performed:** Tester will attempt sign up with an invalid or already used username or email.

##### Verify user account

**Test ID:** FRT1-3

**Type:** Functional, Dynamic, Manual

**Initial State:** After successful user account creation

**Input:** Open verification link in email

**Output:** Success message

**How the test will be performed:** Tester will check account email and open the verification link

### Delete account

**Test ID:** FRT1-4

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated, viewing profile

**Input:** Select delete account, then confirm

**Output:** Deleted account success message

**How the test will be performed:** User will be deleted and then database will be checked and account deletion will be verified

### Log in

**Test ID:** FRT1-5

**Type:** Functional, Dynamic, Manual

**Initial State:** Prompted to log in

**Input:** Enter user information from previously created account

**Output:** Successful or failed authentication **How the test will be performed:** Tester will log in with an email not associated with any account (authentication fails) and then with a previously created user account (authentication succeeds).

### Log in (unverified)

**Test ID:** FRT1-6

**Type:** Functional, Dynamic, Manual

**Initial State:** Prompted to log in

**Input:** Enter user information from previously created account

**Output:** Error message notifying user to verify by email **How the test will be performed:** Tester will log in with a previously created user account. Authentication succeeds but will prompt the user to check email for verification email.

## Log out

**Test ID:** FRT1-7

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated, viewing homepage

**Input:** Log out button selected

**Output:** Log in/Sign Up button shown on homepage

**How the test will be performed:** Log out and then confirm that authenticated user privileges were lost.

## 3.1.2 Content Management

### Post creation

**Test ID:** FRT2-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated, create post modal opened

**Input:** Test post parameters

**Output:** Check post shows in subreddit

**How the test will be performed:** Authenticated account will create a test post and verify that a post shows in the posted subreddit afterwards

### Subreddit creation

**Test ID:** FRT11-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated, create subreddit modal opened

**Input:** Test subreddit name and description

**Output:** Subreddit created

**How the test will be performed:** Authenticated account will create a test subreddit and verify that the subreddit exists and was added to the subreddit navigation list



## Content browsing

**Test ID:** FRT3-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated

**Input:** View homepage and then select a subreddit

**Output:** Show posts from a variety of subreddits on homepage, show posts from specific subreddit when selected

**How the test will be performed:** Tester will manually view and verify that posts from multiple subreddits show on the homepage and when a subreddit is selected, posts from only that specific subreddit is shown

## Edit posts

**Test ID:** FRT4-1

**Type:** Structural, Dynamic, Automated

**Initial State:** Authenticated

**Input:** Send edit test post request to database

**Output:** Test post is modified

**How the test will be performed:** Authenticated test account will send a request to edit a post and then verify that the post has been successfully modified.

## Delete posts

**Test ID:** FRT4-2

**Type:** Structural, Dynamic, Automated

**Initial State:** Authenticated

**Input:** Send delete request for test post to database

**Output:** Test post deleted

**How the test will be performed:** Authenticated test account will send a request to delete a post and then verify that the post has been successfully deleted.

**Edit comment****Test ID:** FRT4-3**Type:** Structural, Dynamic, Automated**Initial State:** Authenticated**Input:** Send edit test comment request to database**Output:** Test comment is modified**How the test will be performed:** Authenticated test account will send a request to edit a comment and then verify that the comment has been successfully modified.**Delete comment****Test ID:** FRT4-4**Type:** Structural, Dynamic, Automated**Initial State:** Authenticated**Input:** Send delete request for test comment to database**Output:** Test comment deleted**How the test will be performed:** Authenticated test account will send a request to delete a comment and then verify that the comment has been successfully deleted.**Create comment****Test ID:** FRT5-1**Type:** Structural, Dynamic, Automated**Initial State:** Authenticated**Input:** Send a request to create a test comment to database**Output:** Test comment created**How the test will be performed:** Authenticated test account will send a request to create a comment and then verify that the comment has been successfully created.

### Up-voting a post

**Test ID:** FRT6-1

**Type:** Structural, Dynamic, Automated

**Initial State:** Authenticated

**Input:** Send a request to upvote test post to database

**Output:** Test post points increased

**How the test will be performed:** Authenticated test account will send a request to upvote a post and then verify that the post points has been successfully incremented.

### Down-voting a post

**Test ID:** FRT6-2

**Type:** Structural, Dynamic, Automated

**Initial State:** Authenticated

**Input:** Send a request to downvote a test post to database

**Output:** Test post points decreased

**How the test will be performed:** Authenticated test account will send a request to downvote a post and then verify that the post points has been successfully decremented.

### Up-voting a comment

**Test ID:** FRT6-3

**Type:** Structural, Dynamic, Automated

**Initial State:** Authenticated

**Input:** Send a request to upvote test comment to database

**Output:** Test comment points increased

**How the test will be performed:** Authenticated test account will send a request to upvote a comment and then verify that the comment points has been successfully incremented.

### Down-voting a comment

**Test ID:** FRT6-4

**Type:** Structural, Dynamic, Automated

**Initial State:** Authenticated

**Input:** Send a request to downvote a test comment to database

**Output:** Test comment points decreased

**How the test will be performed:** Authenticated test account will send a request to downvote a comment and then verify that the comment points has been successfully decremented.

### User karma points

**Test ID:** FRT6-5

**Type:** Structural, Dynamic, Manual

**Initial State:** Authenticated

**Input:** Upvote/downvote a user's post/comment, select the user's name to open user profile

**Output:** User profile modal opened(karma points should have changed)

**How the test will be performed:** Tester will upvote or downvote a user's post or comment and then select the user to view their information. The karma points should then be verified to change according to the upvote or downvote.

### Sorting posts (Home page)

**Test ID:** FRT7-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated, on home page

**Input:** Select all three different sorting options: Hot, New and Top

**Output:** All posts sorted according to selection

**How the test will be performed:** Tester will select different sorting option and verify that sorting works. Hot shows more recently popular posts, New shows newest posts and Top shows posts with highest karma points.

### Sorting posts (Subreddit)

**Test ID:** FRT7-2

**Type:** Functional, Dynamic, Manual

**Initial State:** Authenticated, on test subreddit

**Input:** Select all three different sorting options: Hot, New and Top

**Output:** All posts sorted according to selection

**How the test will be performed:** Tester will select different sorting option on the test subreddit and verify that sorting works. Hot shows more recently popular posts, New shows newest posts and Top shows posts with highest karma points.

### 3.1.3 Navigation Methods

#### View user profile

**Test ID:** FRT8-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Application active

**Input:** Select user to open user profile

**Output:** User profile modal opened

**How the test will be performed:** Tester will select a user to view their information and verify that all of the information is correct with the database.

#### View subreddit

**Test ID:** FRT9-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Application active

**Input:** Select a test subreddit or use subreddit navigation

**Output:** Test subreddit displayed

**How the test will be performed:** Tester will have the application opened and select a subreddit. The subreddit should then be displayed

### Search for existing subreddit

**Test ID:** FRT10-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Application active

**Input:** Input a subreddit name

**Output:** Show subreddit (if it exists)

**How the test will be performed:** Tester will have the application opened and search for a subreddit. Since the subreddit exists, the tester can then verify that the search successfully moves user to specified subreddit.

### Search for non-existing subreddit

**Test ID:** FRT10-2

**Type:** Functional, Dynamic, Manual

**Initial State:** Application active

**Input:** Input a subreddit name

**Output:** No subreddit shown

**How the test will be performed:** Tester will have the application opened and search for a subreddit that does not exist.

## 3.2 Tests for Nonfunctional Requirements

### 3.2.1 Portability

#### Open app on multiple operating systems and platforms

**Test ID:** NFRT1-1

**Type:** Non-Functional, Dynamic, Manual

**Initial State:** Application closed, not running

**Input:** Navigate to website

**Output:** Page will load, posts will load if connected to internet

**How the test will be performed:** Tester will navigate to the application from an internet browser. The home page of the application should load with popular posts.

### 3.2.2 Performance

#### Retrieving data within a 5 second response time

**Test ID:** NFRT2-1

**Type:** Functional, Dynamic, Manual

**Initial State:** Application active

**Input:** Select any button on the application or refreshing the page

**Output:** Page requested loads within 5 seconds.

**How the test will be performed:** Navigation through various pages of the apps and creating posts, logging in, up voting/down voting comments

#### Reliability/Availability: Accessing the application with internet access

**Test ID:** NFRT2-2

**Type:** Functional, Dynamic, Manual

**Initial State:** In a location with wifi access or data access

**Input:** Visit the correct URL of the project hosted

**Output:** Application loads on device

**How the test will be performed:** Tester will be in a location with internet access and try to navigate to the project URL.

### 3.2.3 Security

#### Moderator control

**Test ID:** NFRT3-1

**Type:** Non-Functional, Dynamic, Manual

**Initial State:** Logged on to Firebase on AKT development team account

**Input:** Navigate to authentication database/posts database

**Output:** Specified user accounts/comments/posts deleted

**How the test will be performed:** A member of the AKT development team will access the Firebase databases and remove users or posts/comments/subreddits that contain online harassment or inappropriate content.

**User attempts to access another user's account****Test ID:** NFRT3-2**Type:** Non-Functional, Dynamic, Manual**Initial State:** On website, not logged in as user**Input:** Enter another user's email and some password**Output:** Application should present error in credentials message**How the test will be performed:** A member of the AKT development team will attempt enter a user's email into the login component and a random password, then attempt to log in.

## 4 Comparison to Existing Implementation

In order to ensure that the project has at least the core functions of Reddit, all functional types tests that we use can also be applied to the original Reddit application.

## 5 Unit Testing Plan

### 5.1 Unit testing of internal functions

The Jasmine test framework will be leveraged in order to unit test the functionality of the components and services. The core functions that will be tested are within the database service (`database.service.ts`) which handles almost all of the data transfer between the application and Firebase. These tests will mainly consist of structural tests to verify that the application receives proper information from the database. There will be no use of stubs or drivers in unit testing.