Quantitative Usability Evaluation

April 7th 2017

COMP3008

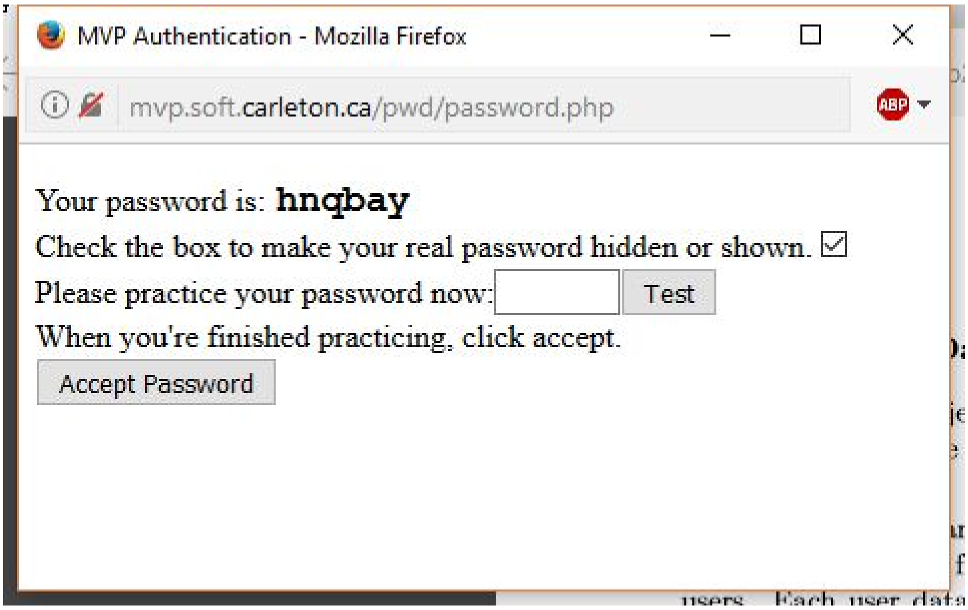
Mike Cichonski, Grayson Lafreniere, Shrey Shree, Shane Slatter

Table of Contents

1. Sample Data and Descriptive Statistics

1.1 Advantages and Disadvantages

The first password scheme is *Text28*. It prompts the user with a randomly generated password of six lowercase alphabetical letters and displays it.


MVP Authentication - Mozilla Firefox
mvp.soft.carleton.ca/pwd/password.php
ABP
Test
Accept Password

Your password is: **hnqbay**

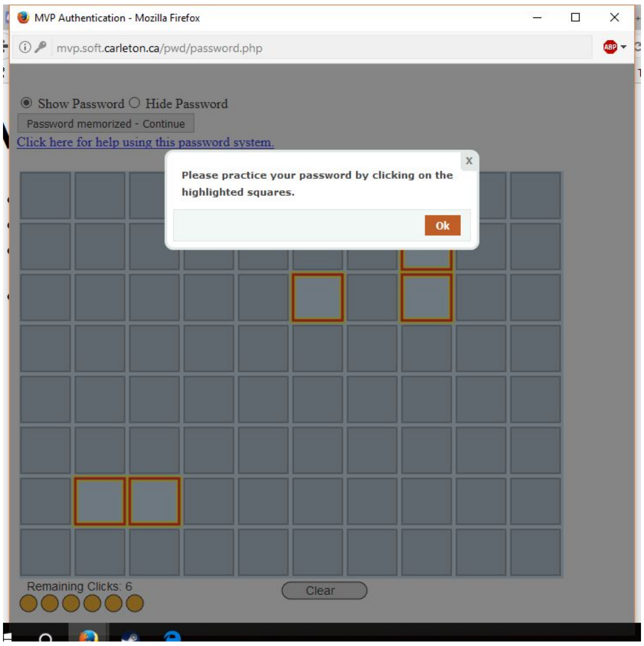
Check the box to make your real password hidden or shown. ☑

Please practice your password now:

When you're finished practicing, click accept.

Figure 1 - *Text28*'s default screen

The second password scheme is *Blankpt28*. It displays an 8x10 grid with six randomly chosen tiles (highlighted) in the grid as the password generated for the user.


MVP Authentication - Mozilla Firefox
mvp.soft.carleton.ca/pwd/password.php
Password memorized - Continue
x
Ok
Clear

● Show Password ○ Hide Password

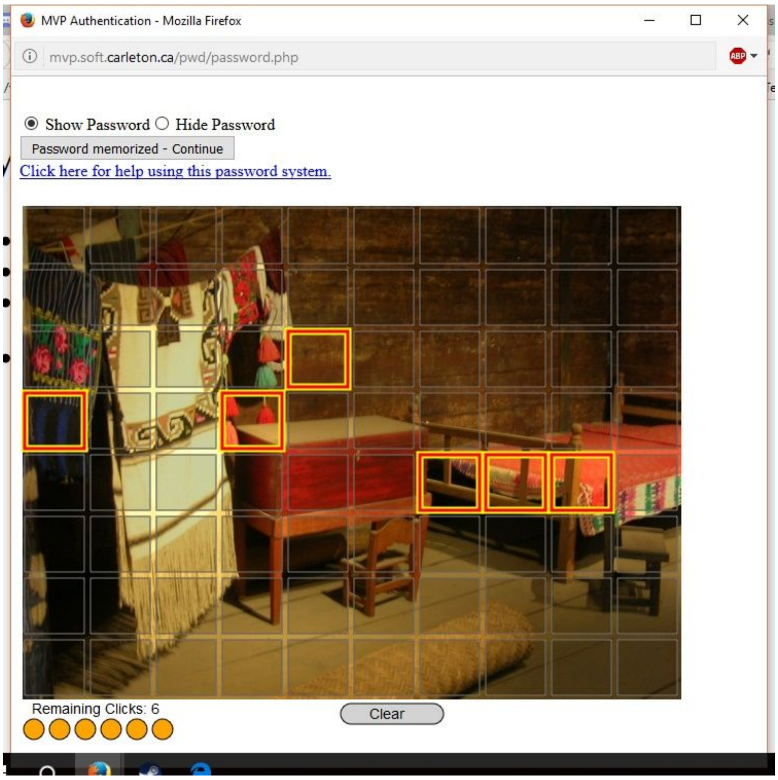
Click here for help using this password system.

Please practice your password by clicking on the highlighted squares.

Remaining Clicks: 6

Figure 2 - *Blankpt28* with default screen

The third password scheme is *Imagept28*. It’s virtually identical to *Blankpt28*, but has an image instead of a blank grid as the background.


MVP Authentication - Mozilla Firefox
mvp.soft.carleton.ca/pwd/password.php
Password memorized - Continue
Click here for help using this password system.
Remaining Clicks: 6
Clear

● Show Password ○ Hide Password

Figure 3 - *Imagept28* with default screen

The three schemes have their appropriate uses. *Text28*, while not actually used in practice, is the most commonly recognized (by regular people) scheme for passwords in the modern world (i.e. most passwords people use are alphanumerical and not fixed to a random permutation of six lowercase characters). That being said, *Text28* is guaranteed a 2^{26} password space, due to the nature of permutations. This password space translates into roughly 2^{28} bits of entropy ($2^{28.202638}$ exact), however a randomly generated password of characters is not easy for the average person to remember. Despite *Text28* only using six characters for the password itself, the characters are not guaranteed to have a coherent pattern; for example, a word in the English language.

Blankpt28 strives to solve the shortcomings of *Text28* by replacing the text-based password approach with a pattern-based approach. The premise of *Blankpt28* uses human being’s natural pattern-recognition skills to reinforce a password by selecting six random tiles in an 8x10 grid. Naturally, the password space of *Blankpt28* is (80 choose ^), translating to approximately 2^{28} bits of entropy ($2^{28.162791}$ exact). However, despite a regular human being’s ability to find patterns in things, Blankpt28 provides no avenues to actually derive a pattern from; it is only eighty tiles with a blank background. It is also not feasible for every password to use an 8x10 grid for the same level of security as *Text28*’s six random-character approach.

Imagept28 is the successor to *Blankpt28*. The difference between *Blankpt28* and *Imagept28* is, as demonstrated in the above figures, *Imagept28* utilizes a randomly-chosen image as the background for the 8x10 grid. Six random tiles on the grid are still chosen, exactly like *Blankpt28*. *Imagept28* boasts the same password space as Blankpt28, but has the added benefit of allowing a person to more easily make patterns to remember the password with the background image. For this reason, *Imagept28* is superior to *Blankpt28*.

Out of the three of these password schemes, people who utilize pattern-based recognition more than straight memorization would prefer *Blankpt28* and/or *Imagept28*. People who prefer a more compact password scheme, and are more receptive to memorizing, would choose *Text28*.

1.2 Documentation

To process the log data we used R. The data given in the files Text28_log.csv, Blankpt28_log.csv and Imagept28_log.csv were organized as timestamps of each event. Each row in the files had the following fields: time, user, site, scheme, mode, event, data and each file was ordered by time. We decided to keep the data sorted by time. This way we could keep track of when a login attempt began and when it finished. We decided the resulting csv files should contain the following information: userid, site, password scheme, event (success or failure), and time taken.

In order to get each login attempt we iterated over the data with a for loop and kept track of each row where the event was “start”. Then we looked for the next row where the event was either “success” or “failure”. We ignored every other event like “create”, “register”, “verifytest” etc… as they had nothing to do with a login attempt. Then we got the time difference between the two events and added a new row to a data frame with all the important information. There were a few cases where the users would start a login and then not complete it and start again. To fix this we had a boolean that made sure there was a “start” event before a “success” or “failure” event. This continued for each successful or failed login attempt. The for loop was created inside a function called “fun” this way we could easily repeat the process for each log data file. For example, fun(text) would get the results for the Text28_log.csv file. There were three resulting data files: TextResult.csv, ImageResult.csv and BlankResult.csv that can be found in the “R” file.

The source code, “function.R”, can be found in the “R” file as well. The source code includes code that uses the resulting data to produce the tables and graphs in the following section 1.3. For just getting the resulting data from the log data is the code from lines 1 to 31 and lines 100-102 for the writing of the resulting csv file. Everything in between lines 31 to 100 is for producing tables and graphs of the resulting data.

1.3 Scheme usability comparison

The source code, “function.R”, can be found in the “R” file. As stated in the above section 1.2, the source code includes code that gets the resulting data from the log data and code that uses the resulting data to produce the tables and graphs. Lines 31 to 100 is for producing tables and graphs of the resulting data. They were both included in the same function to simplify the work. Below you will find the tables and graphs of the resulting data for each password scheme.

Text28_log

Number of logins per user

User	Successful Logins	Failed Logins	Total Logins	Success %
ast403	15	3	18	83.33
ast413	15	0	15	100.00
ast416	14	0	14	100.00
ast417	15	0	15	100.00
ast418	14	2	16	87.50
ast422	15	0	15	100.00
ast423	15	1	16	93.75
ast424	11	0	11	100.00
ast426	15	12	27	55.56

Mean, median and standard deviation for logins

Logins	Mean	Median	Standard Deviation
Success	14.33333	15	1.322876
Failure	2	0	3.905125
Total	16.33333	15	4.415880

Mean Success% = 91.13

Successful login times per user

User	Mean	Median	Standard Deviation
ast403	18.933333	10.0	23.0200741
ast413	5.200000	4.0	2.8334734
ast416	27.571429	20.5	19.2542275
ast417	12.800000	13.0	8.9538499
ast418	6.214286	4.5	3.5121453
ast422	4.400000	3.0	4.1022642
ast423	3.666667	4.0	0.8164966

ast424	9.818182	9.0	5.4372453
ast426	5.466667	5.0	2.5597619

Failed login times per user

User	**Mean**	**Median**	**Standard Deviation**
ast403	6.000000	3.0	6.0827625
ast418	3.500000	3.5	0.7071068
ast423	3.000000	3.0	NA
ast426	5.833333	5.0	2.2087978

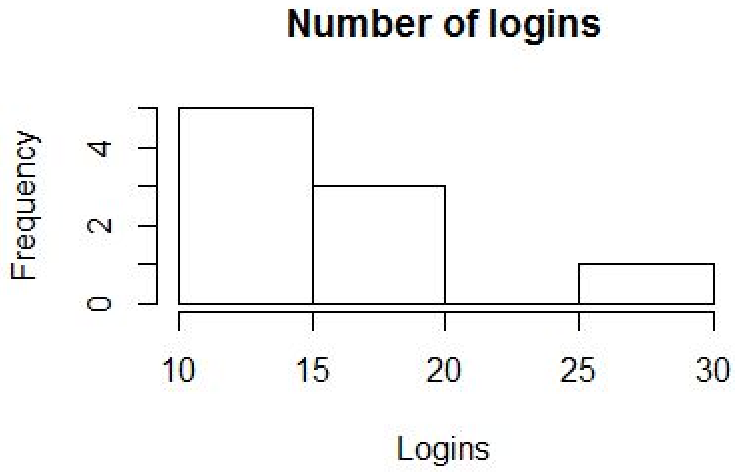


Figure 4 - Histogram for total number of logins (Text28 scheme)

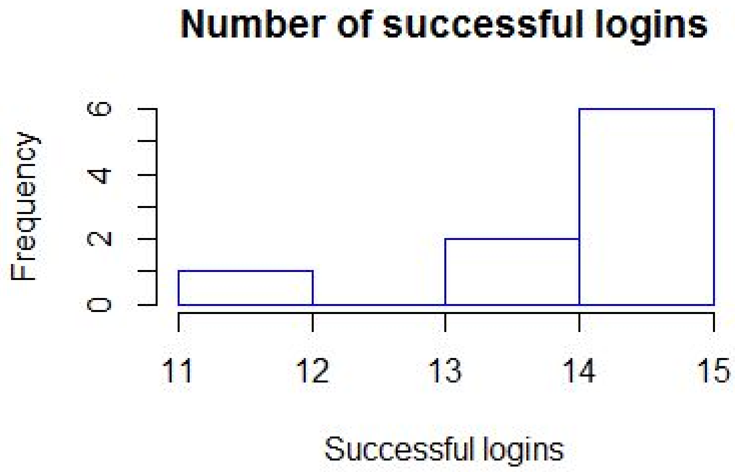


Figure 5 - Histogram for number of successful logins (Text28 scheme)

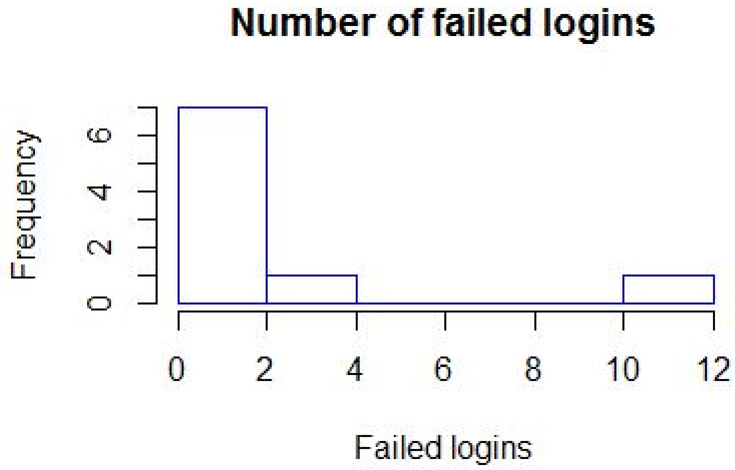


Figure 6 - Histogram for number of failed logins (Text28 scheme)

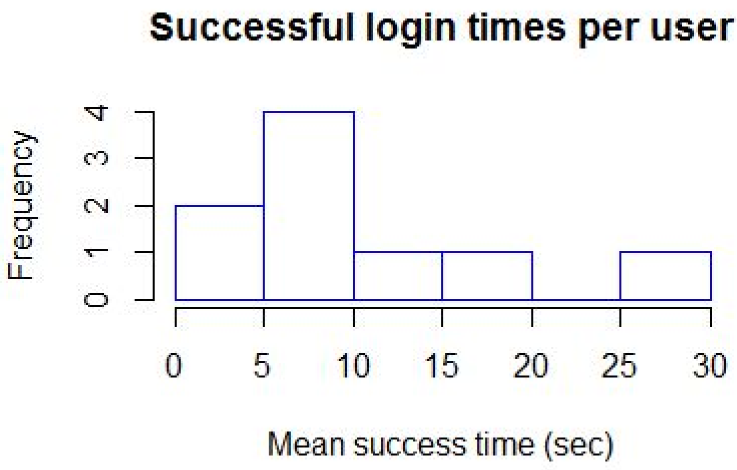


Figure 7 - Histogram for successful login times (Text28 scheme)

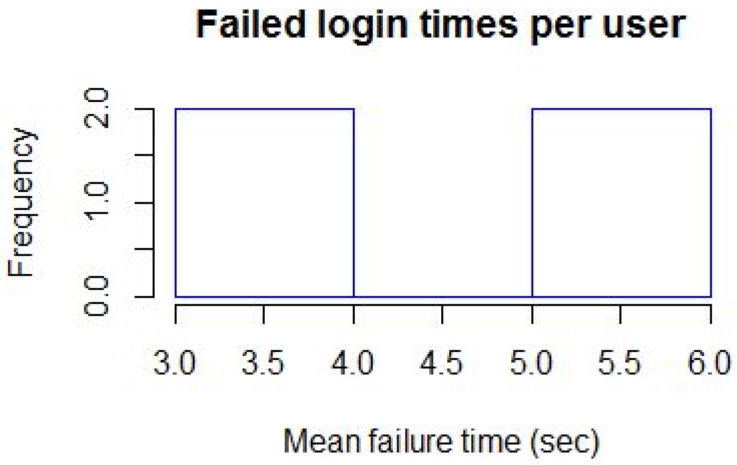


Figure 8 - Histogram for failed login times (Text28 scheme)

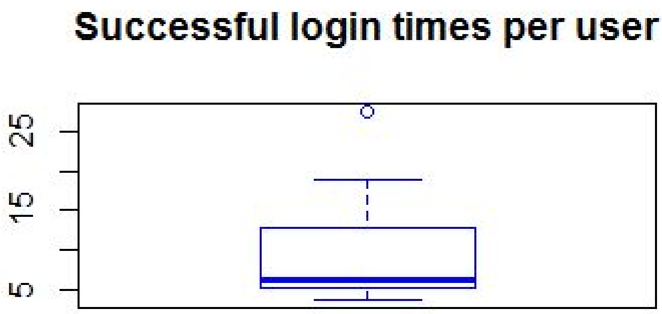


Figure 9 - Boxplot for successful login times (Text28 scheme)

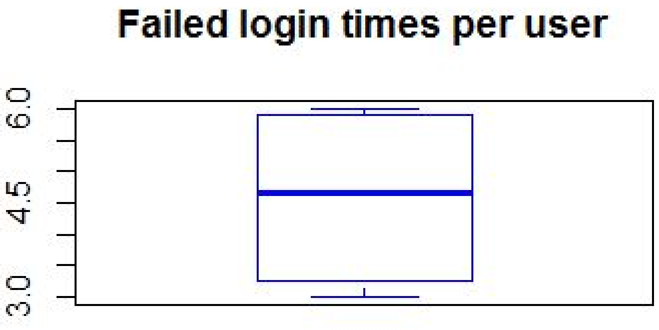


Figure 10 - Boxplot for failed login times (Text28 scheme)

Interpretation of the Text28 scheme statistics and graphs

In the above table *Number of logins per user* we can conclude the following:

- There were 9 users total.
- More than half of the users got 100% successful login rate.
- Only one person failed more than 3 times with 12 failed login attempts.

The graphs that represent this table are *figure 4, figure 5, figure 6*. From the graphs we can conclude the following:

- Most users logged in a total of 10-15 times.
- Most users successfully logged in 14-15 times.
- Most users failed the login 0-2 times.

In the above table *Mean, median and standard deviation for logins* we can conclude the following:

- The mean total login attempts is 16.
- The standard deviation for successful logins is 1.33 meaning the resulting number of successful logins are consistent between users.
- The standard deviation for failed logins is 3.90 meaning the resulting number of failed logins are not as consistent between failed logins.
- The standard deviation for total logins is 4.41 meaning some users logged in more or less than the mean total login amount.
- The average user succeeded logins 91.13% of the time.

In the above table *Successful login times per user* we can conclude the following:

- Only 3 out of 9 users had a standard deviation below 3.
- The rest were between 3 and 23 meaning the success login times were not consistent.

In graphs that represent this table are *figure 7* and *figure 9*. From these graphs we can conclude that it takes the average user between 5-10 seconds to login successfully.

In the above table *Failed login times per user* we can conclude the following:

- Only 4 out of the 9 total users failed an attempt so the statistics are lacking.

- There were a total of 18 total failed login attempts where 12 came from a single user because of this the data is useless.

The graphs that represent this table are *figure 8* and *figure 10*. From these graphs we can conclude that it takes the average user between 3-4 and 5-6 seconds for failed logins. The times should be similar to the successful login times. This seems like failed login attempts were because the user did not know or even tried to guess and tried to login without typing much of a password.

Blankpt28_log

Number of logins per user

User	Successful Logins	Failed Logins	Total Logins	Success %
bpt409	15	3	18	83.33
bpt411	15	7	22	68.18
bpt416	15	1	16	93.75
bpt418	15	2	17	88.24
bpt419	13	0	13	100.00
bpt422	13	5	18	72.22
bpt423	12	5	17	70.59
bpt424	9	1	10	90.00
bpt425	12	0	12	100.00

Mean, median and standard deviation for logins

Logins	Mean	Median	Standard Deviation
Success	13.22222	13	2.048034
Failure	2.666667	2	2.5
Total	15.88889	17	3.655285

Mean Success% = 85.15

Successful login times per user

User	Mean	Median	Standard Deviation
bpt409	9.733333	8	6.485882
bpt411	14.533333	14	5.396648

bpt416	37.333333	37	6.343350
bpt418	16.200000	15	4.916445
bpt419	38.923077	34	14.648444
bpt422	34.461538	30	15.180115
bpt423	27.583333	26	6.444989
bpt424	26.222222	25	7.327649
bpt425	27.416667	20	19.265883

Failure login times per user

User	**Mean**	**Median**	**Standard Deviation**
bpt409	13.00000	9.0	6.928203
bpt411	25.28571	29.0	16.610381
bpt416	54.00000	54.0	NA
bpt418	15.50000	15.5	12.020815
bpt422	51.80000	63.0	22.264321
bpt423	39.00000	30.0	21.644861
bpt424	29.00000	29.0	NA

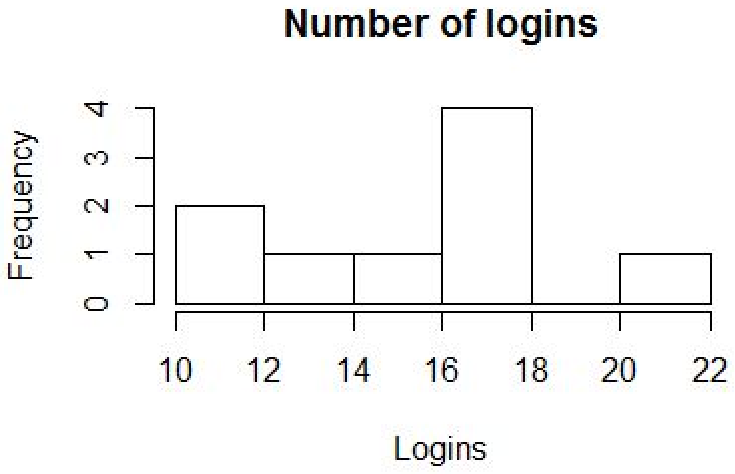


Figure 11 - Histogram for total number of logins (Blankpt28 scheme)

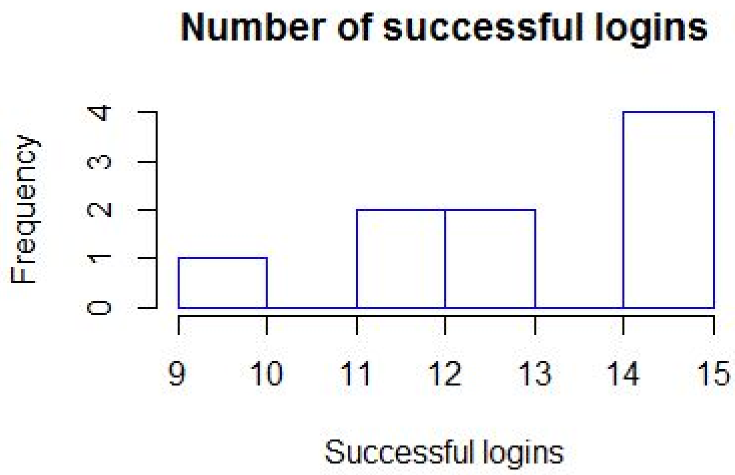


Figure 12 - Histogram for number of successful logins (Blankpt28 scheme)

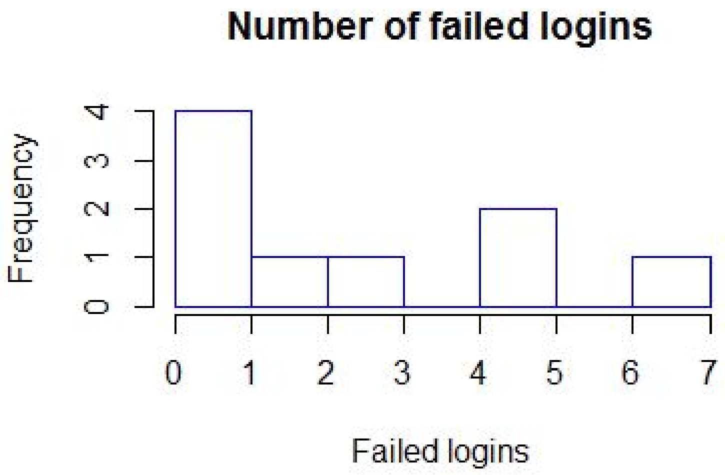


Figure 13 - Histogram for number of failed logins (Blankpt28 scheme)

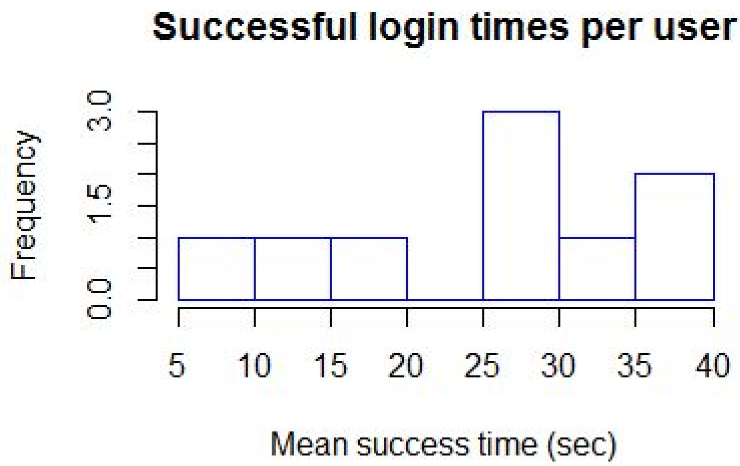


Figure 14 - Histogram for successful login times (Blankpt28 scheme)

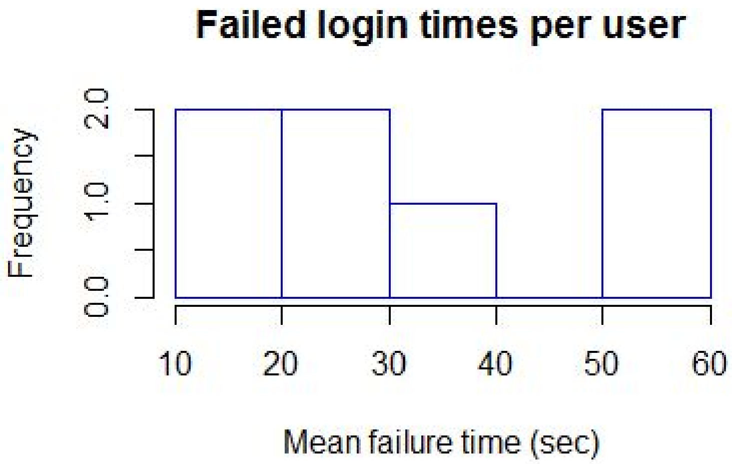


Figure 15 - Histogram for failed login times (Blankpt28 scheme)

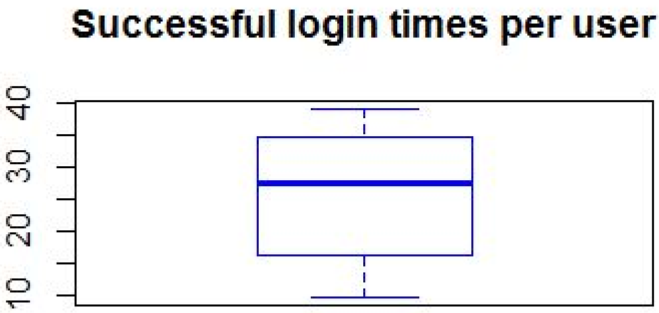


Figure 16 - Boxplot for successful login times (Blankpt28 scheme)

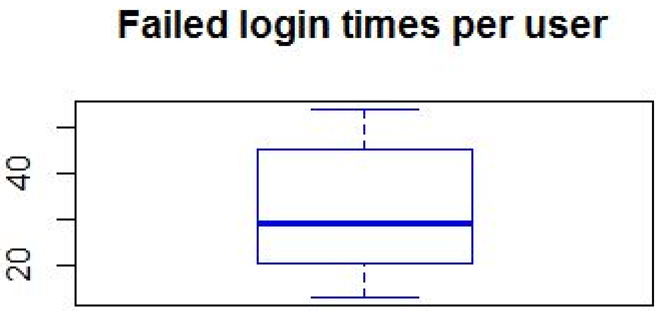


Figure 17 - Boxplot for failed login times (Blankpt28 scheme)

Interpretation of the Blankpt28 scheme statistics and graphs

In the above table *Number of logins per user* we can conclude the following:

- There were 9 users total.
- Only two users succeeded 100% of the time
- Only one user succeeded less than 70% of the time.

The graphs that represent this table are *figure 11, figure 12, figure 13*. From the graphs we can conclude the following:

- Most users logged in a total of 16-18 times.
- Most users successfully logged in 14-15 times.
- Most users failed the login 0-1 times.

In the above table *Mean, median and standard deviation for logins* we can conclude the following:

- The mean total login attempts is 15.89.
- The standard deviation for successful logins is 2.04 meaning the resulting number of successful logins are consistent between users.
- The standard deviation for failed logins is 2.5 meaning the resulting number of failed logins are not as consistent between failed logins.
- The standard deviation for total logins is 3.366 meaning some users logged in more or less than the mean total login amount.
- The average user succeeded logins 85.15% of the time.

In the above table *Successful login times per user* we can conclude the following:

- Only one users had a standard deviation below 5 with 4.9.
- The rest were above 5 meaning the success login times were not consistent.

In graphs that represent this table are *figure 14* and *figure 16*. From these graphs we can conclude that it takes the average user between 25-30 seconds to login successfully.

In the above table *Failed login times per user* we can conclude the following:

- Only 2 out of the 9 total users did not failed an attempt.
- Only one standard deviation was below 10, with 6.93 the rest were above 10 with the highest being 22.26. This means the failure times are very different between attempts for the same user.

The graphs that represent this table are *figure 15* and *figure 17*. From these graphs we can conclude that it takes the average user between 10-30 and 50-60 seconds for failed logins. The times should be similar to the successful login times. This shows that the failure times vary between users drastically so the data is useless.

Imagept28_log

Number of logins per user

<u>User</u>	<u>Successful Logins</u>	<u>Failed Logins</u>	<u>Total Logins</u>	<u>Success %</u>
ipt401	12	0	12	100.00

ipt402	15	0	15	100.00
ipt403	15	5	20	75.00
ipt404	17	6	23	73.91
ipt406	15	7	22	68.18
ipt411	15	4	19	78.95
ipt413	20	6	26	76.92
ipt414	14	2	16	87.50
ipt419	15	4	19	78.95
ipt423	14	1	15	93.33
ipt424	15	2	17	88.24
ipt426	15	2	17	88.24
ipt427	15	4	19	78.95
ipt431	12	5	17	70.59
ipt432	15	1	16	93.75
ipt433	15	0	15	100.00
ipt434	18	2	20	90.00
ipt439	13	1	14	92.86
ipt441	16	7	23	69.57
ipt443	16	1	17	94.12
ipt444	15	1	16	93.75
ipt446	17	2	19	89.47
ipt448	13	1	14	92.86
ipt449	14	0	14	100.00

Mean, median and standard deviation for logins

Logins	**Mean**	**Median**	**Standard Deviation**
Success	15.04167	15	1.781039
Failure	2.666667	5	2.315668

Total	17.70833	17	3.406888

Mean Success% = 86.46

Successful login times per user

User	**Mean**	**Median**	**Standard Deviation**
ipt401	27.25000	23.5	9.449627
ipt402	20.53333	21.0	8.348367
ipt403	21.93333	22.0	4.199773
ipt404	24.41176	18.0	16.699771
ipt406	26.00000	25.0	6.834785
ipt411	28.86667	29.0	3.814758
ipt413	37.90000	37.0	16.657225
ipt414	21.71429	20.0	11.438320
ipt419	24.53333	23.0	8.484159
ipt423	32.57143	29.5	6.559952
ipt424	52.40000	52.0	23.079985
ipt426	37.93333	35.0	17.559965
ipt427	19.86667	16.0	14.065493
ipt431	27.83333	26.5	10.355792
ipt432	31.06667	32.0	12.919900
ipt433	24.73333	22.0	16.117722
ipt434	29.44444	27.0	13.214173
ipt439	25.23077	23.0	10.528959
ipt441	33.06250	35.0	11.946931
ipt443	35.50000	35.0	7.805981
ipt444	12.40000	12.0	2.229670
ipt446	48.23529	47.0	19.829049
ipt448	33.84615	29.0	21.118105

ipt449	37.57143	33.5	15.330785

Failure login times per user

User	Mean	Median	Standard Deviation
ipt403	20.40000	19.0	3.646917
ipt404	27.00000	26.0	8.124038
ipt406	33.85714	31.0	16.476535
ipt411	44.50000	43.0	13.304135
ipt413	32.00000	32.5	11.099550
ipt414	34.00000	34.0	8.485281
ipt419	33.00000	31.0	6.928203
ipt423	4.00000	4.0	NA
ipt424	26.00000	26.0	8.485281
ipt426	42.50000	42.5	16.263456
ipt427	37.50000	36.0	11.818065
ipt431	45.40000	43.0	11.970798
ipt432	35.00000	35.0	NA
ipt434	49.00000	49.0	41.012193
ipt439	59.00000	59.0	NA
ipt441	36.28571	28.0	25.375279
ipt443	32.00000	32.0	NA
ipt444	24.00000	24.0	NA
ipt446	34.50000	34.5	9.192388
ipt448	31.00000	31.0	NA

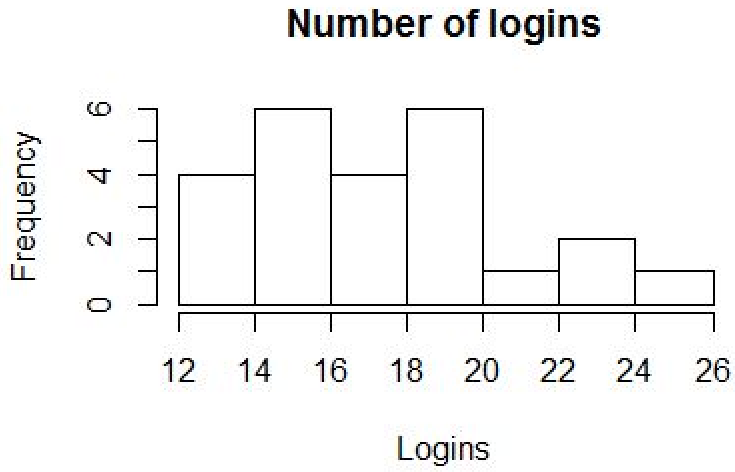


Figure 18 - Histogram for total number of logins (Imagept28 scheme)

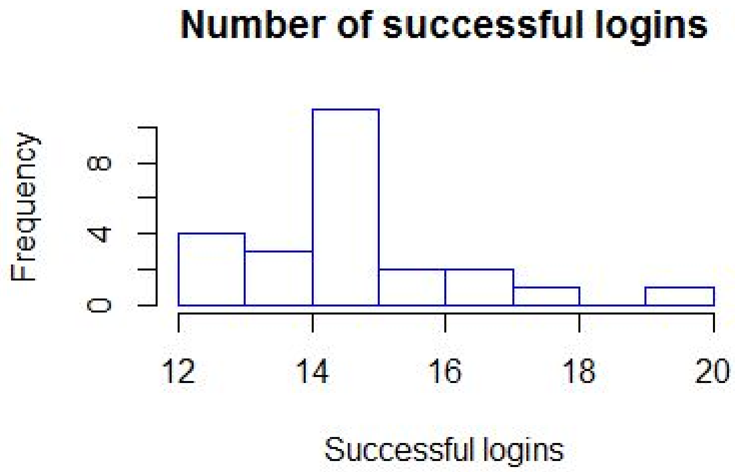


Figure 19 - Histogram for number of successful logins (Imagept28 scheme)

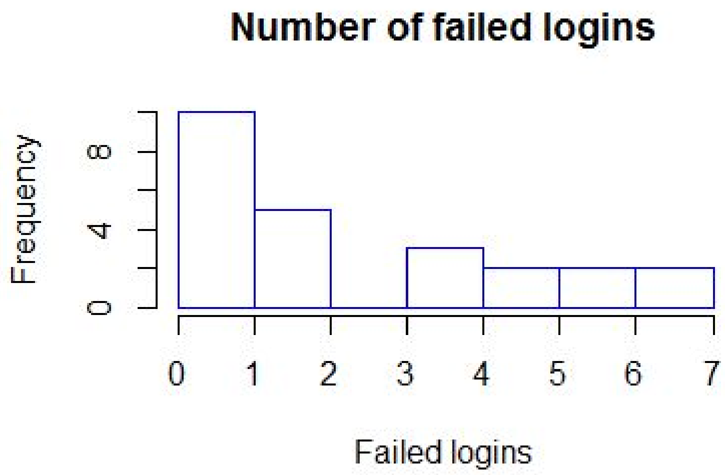


Figure 20 - Histogram for number of failed logins (Imagept28 scheme)

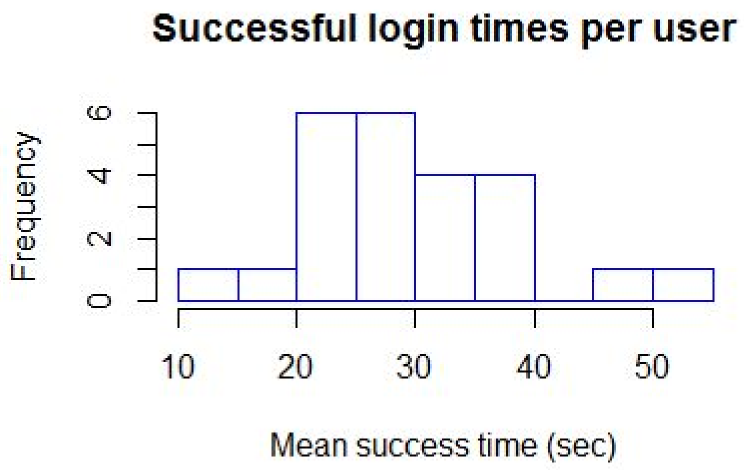


Figure 21 - Histogram for successful login times (Imagept28 scheme)

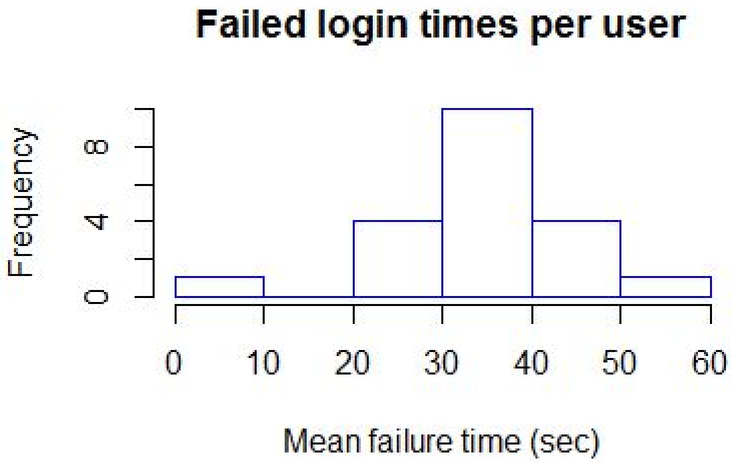


Figure 22 - Histogram for failed login times (Imagept28 scheme)

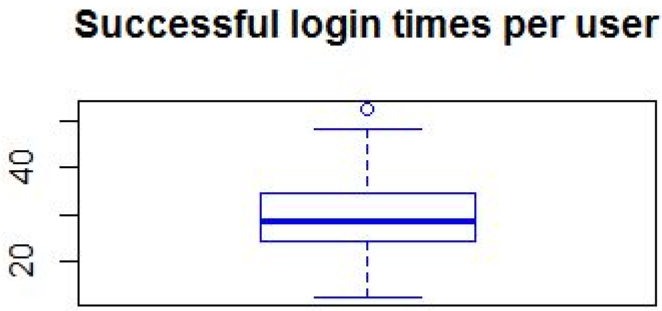


Figure 23 - Boxplot for successful login times (Imagept28 scheme)

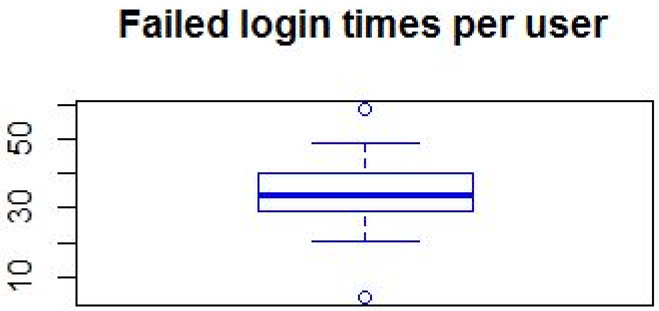


Figure 24 - Boxplot for failed login times (Imagept28 scheme)

Interpretation of the Imagept28 scheme statistics and graphs

In the above table *Number of logins per user* we can conclude the following:

- There were 24 users total.
- Only four users succeeded 100% of the time.
- Only 4 user failed more than 5 times.

The graphs that represent this table are *figure 18, figure 19, figure 20*. From the graphs we can conclude the following:

- Most users logged in a total of 14-16 and 18-20 times.
- Most users successfully logged in 14-15 times.
- Most users failed the login 0-1 times.

In the above table *Mean, median and standard deviation for logins* we can conclude the following:

- The mean total login attempts is 17.71.
- The standard deviation for successful logins is 1.78 meaning the resulting number of successful logins are consistent between users.
- The standard deviation for failed logins is 2.32 meaning the resulting number of failed logins are not as consistent between failed logins.
- The standard deviation for total logins is 3.41 meaning some users logged in more or less than the mean total login amount.
- The average user succeeded logins 86.46% of the time.

In the above table *Successful login times per user* we can conclude the following:

- Only three users had a standard deviation below 5.
- The rest were between 5 and 23 meaning the success login times were not consistent.

In graphs that represent this table are *figure 21* and *figure 23*. From these graphs we can conclude that it takes the average user between 20-30 seconds to login successfully.

In the above table *Failed login times per user* we can conclude the following:

- Only 4 out of the 24 total users did not failed an attempt.
- There were no sd below 5 and the highest sd was 41 meaning all failed login times were very inconsistent.

The graphs that represent this table are *figure 22* and *figure 24*. From these graphs we can conclude that it takes the average user between 30-40 seconds for failed logins. The times are a little longer than the successful login times. This makes sense because the user would take more time trying to remember instead of just inputting the password.

Conclusion

To conclude we will be looking at both success rate and time taken for each scheme to determine the one with the best usability. We believe these two password properties are what makes a password usable.

For the success rate, the Text28 scheme has a successful login rate of 91.13% however there was an outlier with 12 failed logins when the next users highest failed login amount was 3. If we were to remove this outlier the successful login rate would increase.

The Blankpt28 scheme had a successful login rate of 85.15%. The data seemed very consistent as the standard deviation for failed and total login attempts was much lower than the Text28 schemes. There was no clear outlier so we conclude this login rate is correct. For the Imagept28 scheme the successful login rate was 86.46%. Again the standard deviation for failed and total login attempts was much lower than the Text28 schemes. So we can conclude that this data is correct with little error. The successful login rates were 85%(Blankpt28), 86%(Imagept28) and 91%(Text28) which are all very close. However, since there is the obvious outlier in Text28, the percentage would increase if removed. So we conclude that the Text28 has the best successful login rate. This may be due to the fact that Text28 is a common password scheme that all users are very familiar with. The other two schemes may have their success rates increased if users were more familiar with those schemes.

For the time taken to enter the password scheme, the Text28 scheme took an average of 5-10 seconds to login successfully and 3-6 seconds for failed login attempts. With this information we concluded that the failed login times could not have been real login attempts as the times should have been longer because the user would have to think about the password then type the password. The Blankpt28 scheme took an average of 25-30 seconds to login successfully and 10-30 and 50-60 seconds for failed login attempts. Since the failed login times The Imagept28 scheme took an average of 20-30 seconds to login successfully and 30-40 seconds for failed login attempts. With this information we conclude that the failed login times for Text28 and Blankpt28 are useless as they are scattered and too low for a real failed login attempt. The only failed login attempt time that makes sence is Imagept28. However, the fastest time taken to enter the password scheme is Text28. It is 3 times faster than the other two schemes. Again this is probably due to the fact that users are familiar with this password scheme and typing on a keyboard is faster than clicking squares.

To conclude, Text28 scheme has the highest success rate and the fastest login times. We concluded that this is probably due to the fact that all users are familiar with this password scheme. If in the future Blankpt28 and Imagept28 schemes are used more, users may be able to remember their passwords better and enter them faster but currently Text28 is still the best usable password scheme.

2. Design, Implementation, Statistical Inference

2.1 The Scheme

With our password scheme we attempted to accommodate for users' differing learning styles. Some might be better visual learners while others might be better with numbers so we created a password scheme that allows for different memorization techniques. It makes use of both visual and text-based cues to help with password memorization, which engages multiple areas of the brain, further aiding the memorization process.

The biggest drawback to this scheme is a possibility of shoulder surfing, although it appears to be quite difficult to remember the locations of the circles after just one viewing which is why we gave the user three rounds of training for each password. Although this does pose some risk to security, in almost all cases this drawback would be greatly outweighed by the usability aspects of the scheme.

The password scheme has a very simple interface with very constrained interaction (the circles appear to be the only thing that can be 'manipulated' in any way), which makes it obvious very quickly how the scheme works. It is also similar to other grid-based schemes and even visually comparable to the recently standardized 9-dot phone patterns which makes learnability almost instantaneous. The interaction also works smoothly with messages guiding the user through every step. Due to the visual on-screen nature of the scheme, it is ideal for touch screen based machines such as phones and tablets, and it appeared to be faster for users who used the touch screen versus those who clicked and dragged the mouse.

Although the password space of the Phone password (4 values in a 4x4 grid space) does not reach 2^28, the grid size can easily be adjusted to greatly expand the space and meet that requirement. A 4-value password would require a 14x14 grid to meet the 2^28 requirement, which is only a few squares larger than the Bank password space (all calculations below). The idea is that expanding grid size only slightly increases the difficulty of remembering the password (at least numerically) as long as the number of circle values to remember remains constant. We settled for a 4-number password because this can still easily be stored in working memory while allowing for a theoretically large enough space. A 5-number password might also be a reasonable option, but any more than that would make it exceedingly difficult to remember because the values are numbers, not just digits (0-9). Any less than 4 would make the password space too small, but still might be reasonable for a phone password. The values may also be duplicates (the circles can overlap onto the same square) which increases the password space as well. Below is a table of calculations for different grid sizes of the password space given 4 circles. Given that duplicates are allowed, the calculations are very simple.

Grid size	Password space	Total Possibilities	2^28
4x4	4^4	256	268,435,456
7x7	4^7	16,384	268,435,456
10x10	4^10	1,048,576	268,435,456
12x12	4^12	16,777,216	268,435,456

14x14	4^14	268,435,456	268,435,456

2.2 and 2.3 Implementation V1 & V2

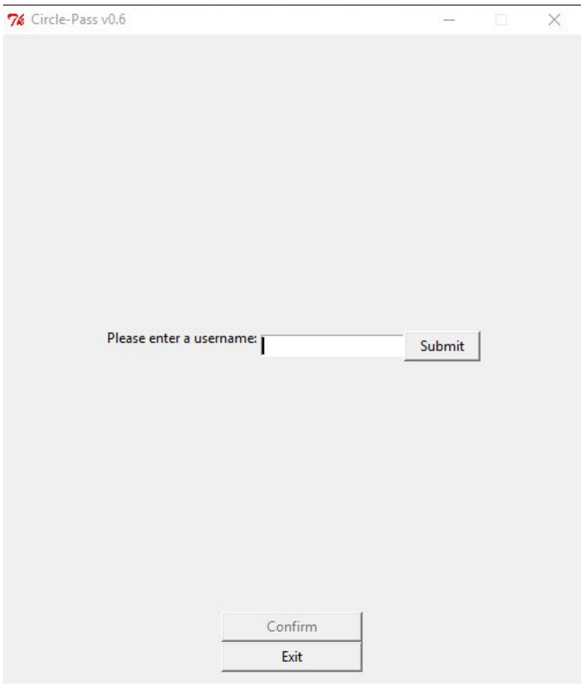
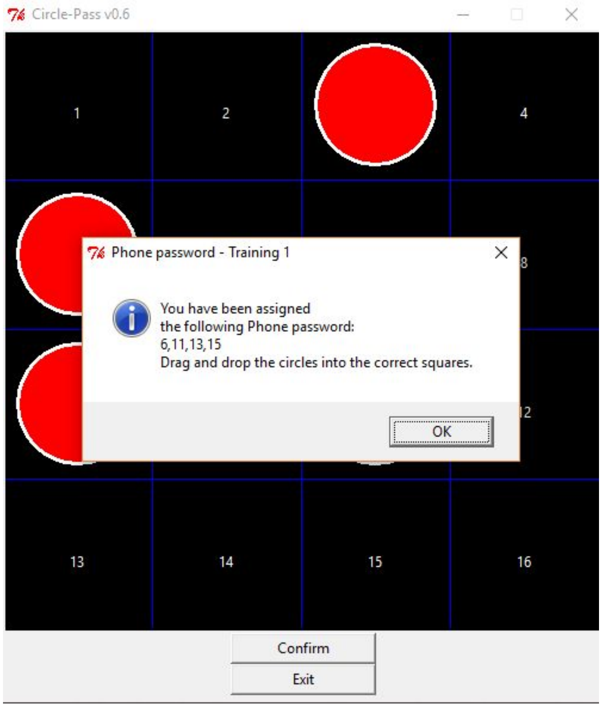


Figure 25 - Default Screen of *CirclePass*

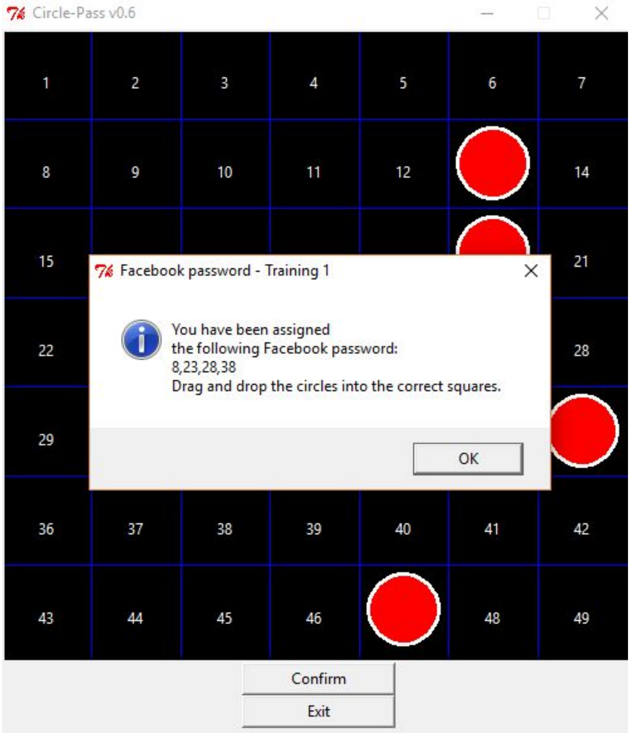
Our revised implementation first generated a “login” screen to help identify specific users in the stats.csv file. After inputting a user name, the program displays the “Phone Password,” a randomly-chosen four tile password in a 4x4 grid.



You have been assigned
the following Phone password:
6,11,13,15
Drag and drop the circles into the correct squares.

Figure 26 - Phone password of *CirclePass*

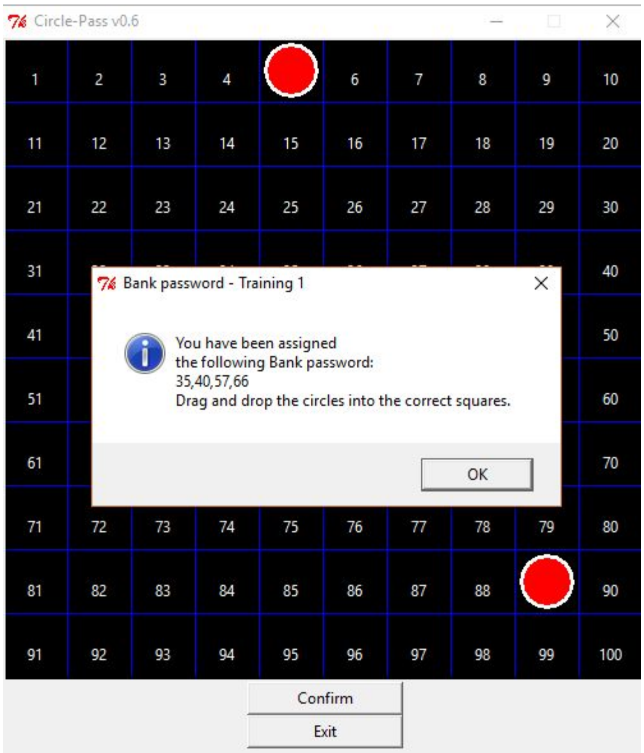
The user is prompted to slide the tiles to the correct locations (i.e. entering in the password). After the first successful entry, the user is asked to re-enter the password another two more times to help memorize the password. If the user enters the wrong password at any point in this stage, a warning will display indicating: the incorrect password was entered, and the correct password will be shown again.



You have been assigned
the following Facebook password:
8,23,28,38
Drag and drop the circles into the correct squares.

Figure 27 - Facebook password for *CirclePass*

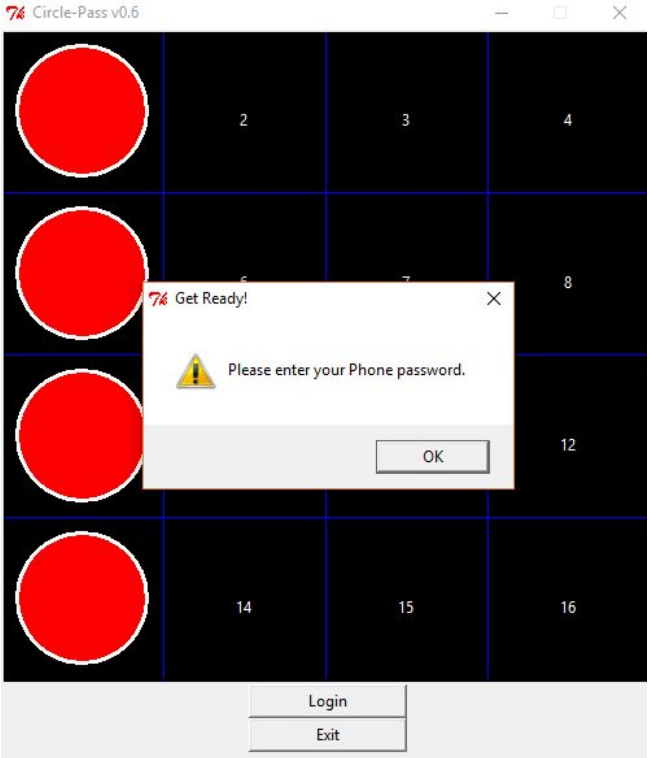
After successfully completing the training for the phone password, second dialog appears with the user's randomly-generated password. For the facebook, it is four randomly-chosen tiles in a 7x7 grid. Much like the Phone password, the user is prompted to input the correct password three times as part of the training phase.



You have been assigned
the following Bank password:
35,40,57,66
Drag and drop the circles into the correct squares.

Figure 28 - Bank password for *CirclePass*

Much like the transition from the Phone password to the Facebook password, once the user successfully enters in the Facebook password three times in the training phase, the last password -- Bank password -- is displayed to the user. Similar to Phone and Facebook password, Bank password is four randomly-chosen tiles in a 10x10 grid. After a user has input the Bank password successfully three times, the training phase will end. The testing phase begins immediately after.

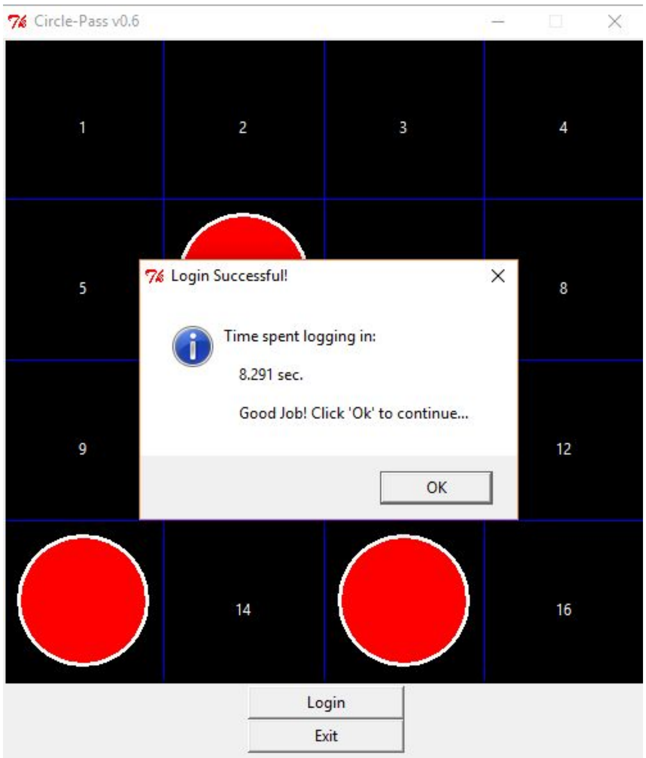


Get Ready!

Please enter your Phone password.

Figure 29 - Testing phase for Phone password

The user is prompted to input his/her assigned password for one of the three services (Phone, Facebook, Bank) at random. In this case, Phone was simply chosen first.



Login Successful!

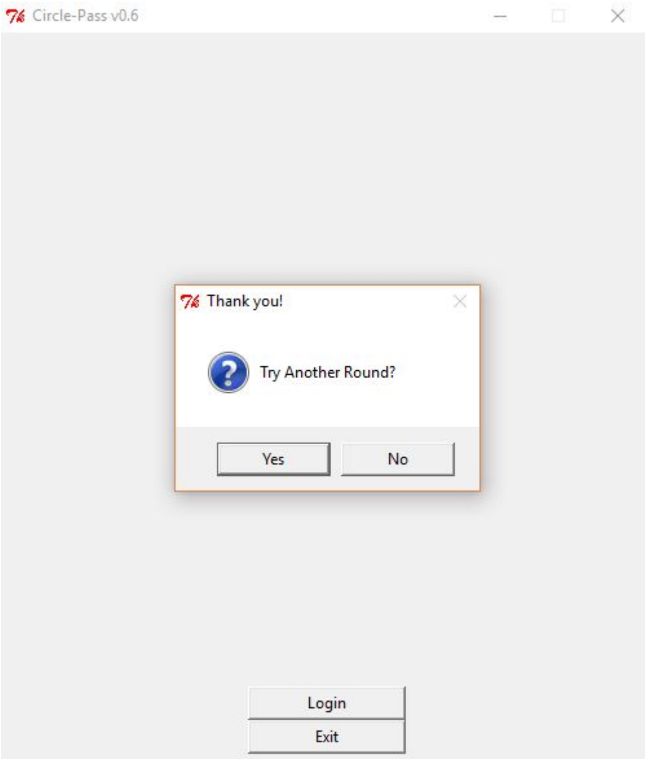
Time spent logging in:

8.291 sec.

Good Job! Click 'Ok' to continue...

Figure 30 - Successful entry in the Testing phase

When the user inputs the correct password (a maximum of three attempts are permitted before moving to the next password), the time taken to login is displayed. This process repeats for the remaining two passwords.



Circle-Pass v0.6

Thank you!

Try Another Round?

Figure 31 - *CirclePass* has finished

After all three passwords have been entered (or attempted) in the Testing phase, the current iteration of *CirclePass* will end. All of the information regarding the current iteration (i.e. passwords, login time, number of attempts, success/failure) are recorded in the stats.csv file under the user's chosen username in the default screen. The program then prompts the user for another round, which if desired will assign new passwords and restart the training phase. If denied, the program will exit.

2.4 Survey

The link to the Password Survey:
https://hotsoft.carleton.ca/comp3008limesurvey/index.php/866772?lang=en

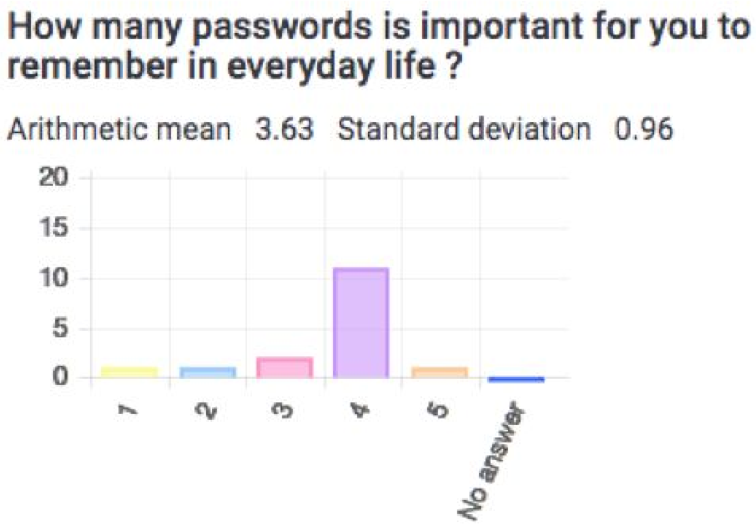
The PDF version of the survey questions is included in the project folder.

The survey was implemented using the Lime Survey. Lime Survey is an online application that is used to generate surveys online. The software generates a link to the survey, which can be emailed to the participants to get feedback. Participants can do the survey online by just opening the link that is sent them. Once the survey is completed the data is automatically updated in the survey software. Lime Server keeps track of the feedback in a statistical graph of each question. The graphs can be found in the responses tab in the Lime Survey.

For the password Scheme software, there were 15 survey questions that were implemented. There were 15 participants who tested our circlepass program. The feedback of the program was evaluated through the survey. .

Graphs below show the results of the survey. The graph includes the standard deviation and arithmetic mean for some feedback question.

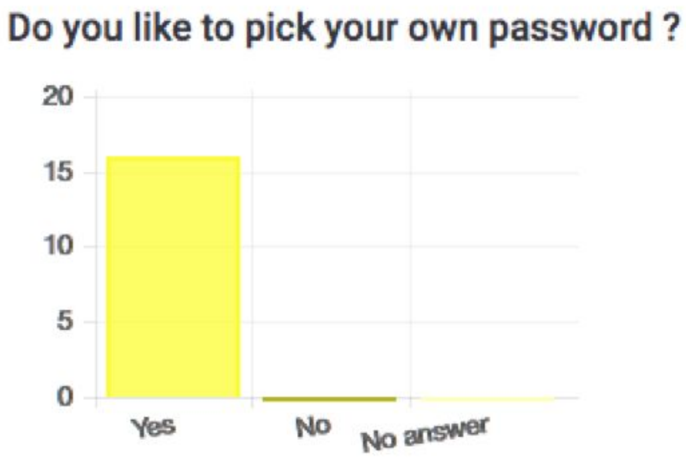
Question 1



How many passwords is important for you to remember in everyday life ?

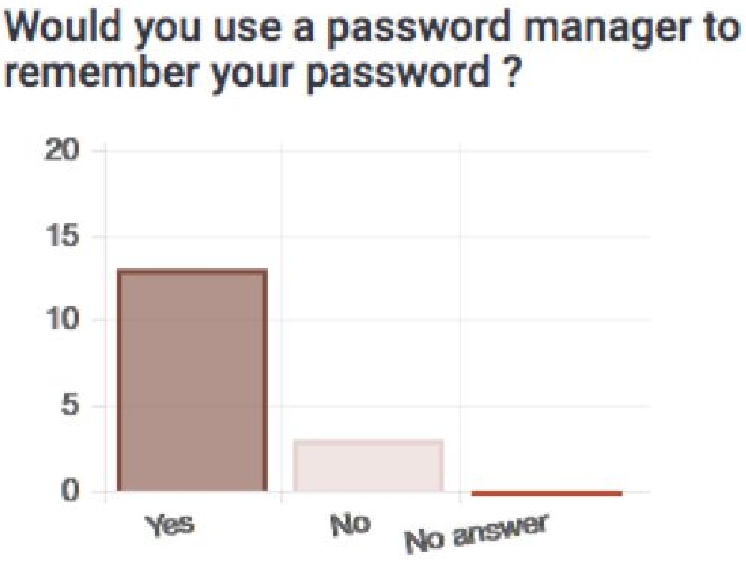
Arithmetic mean 3.63 Standard deviation 0.96

Question 2

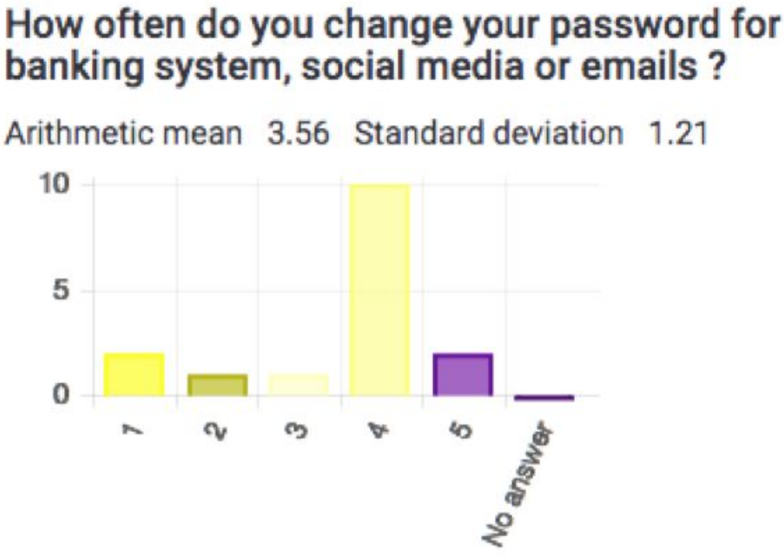


Do you like to pick your own password ?

Question 3

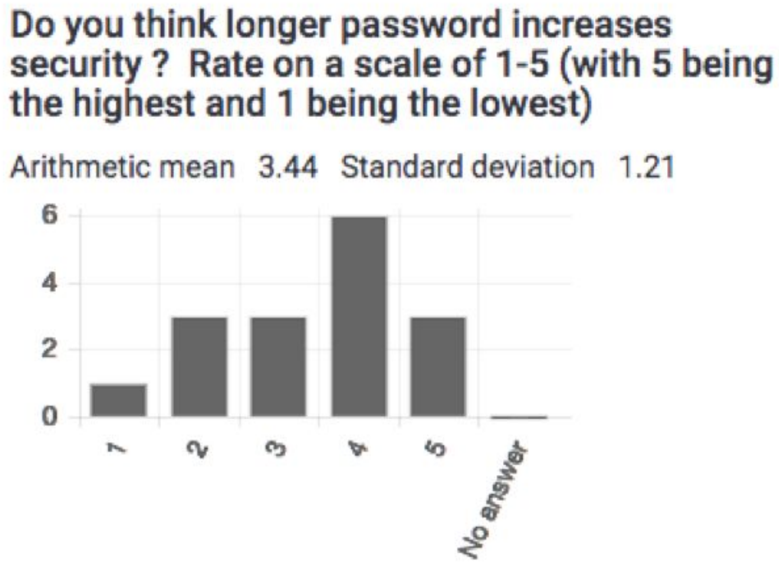

Would you use a password manager to remember your password ?

Question 4


How often do you change your password for banking system, social media or emails ?

Arithmetic mean 3.56 Standard deviation 1.21

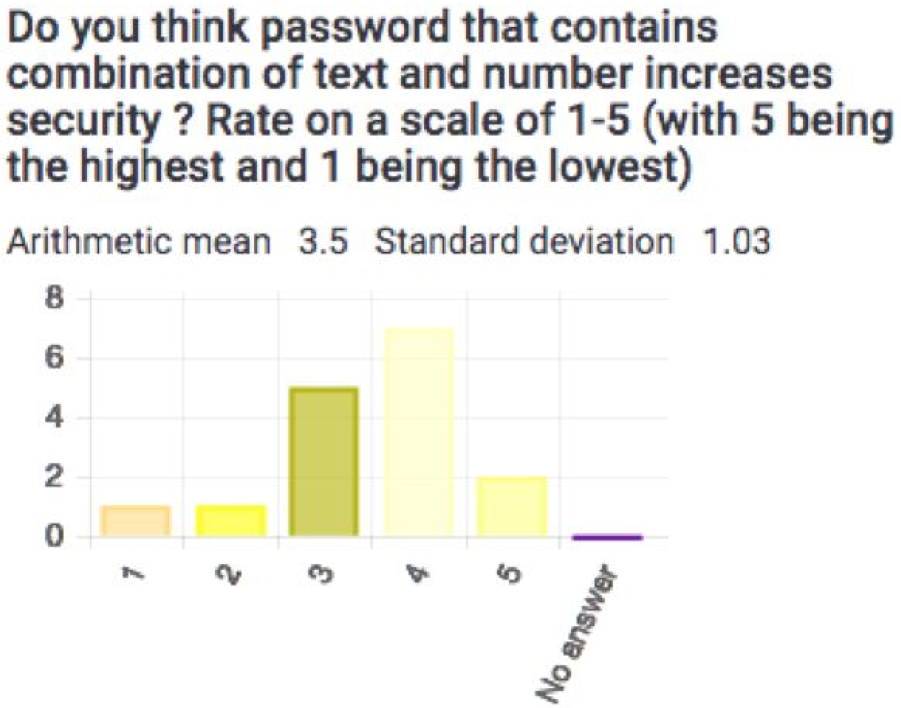
Question 5



Do you think longer password increases security ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

Arithmetic mean 3.44 Standard deviation 1.21

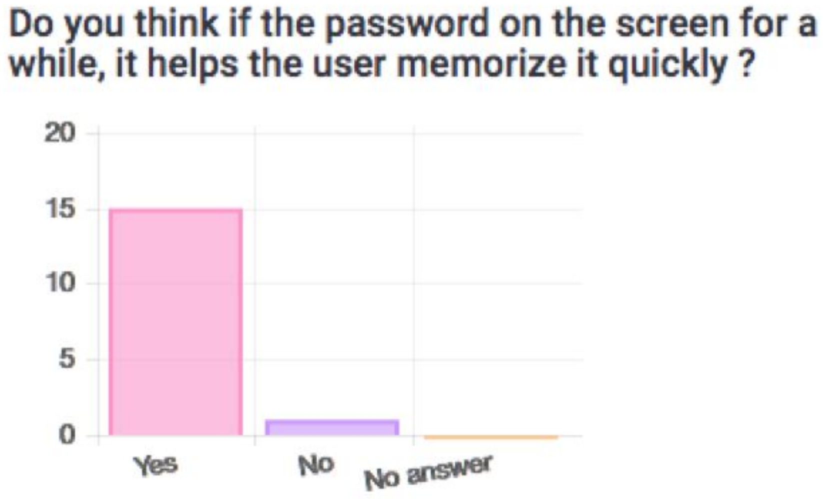
Question 6



Do you think password that contains combination of text and number increases security ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

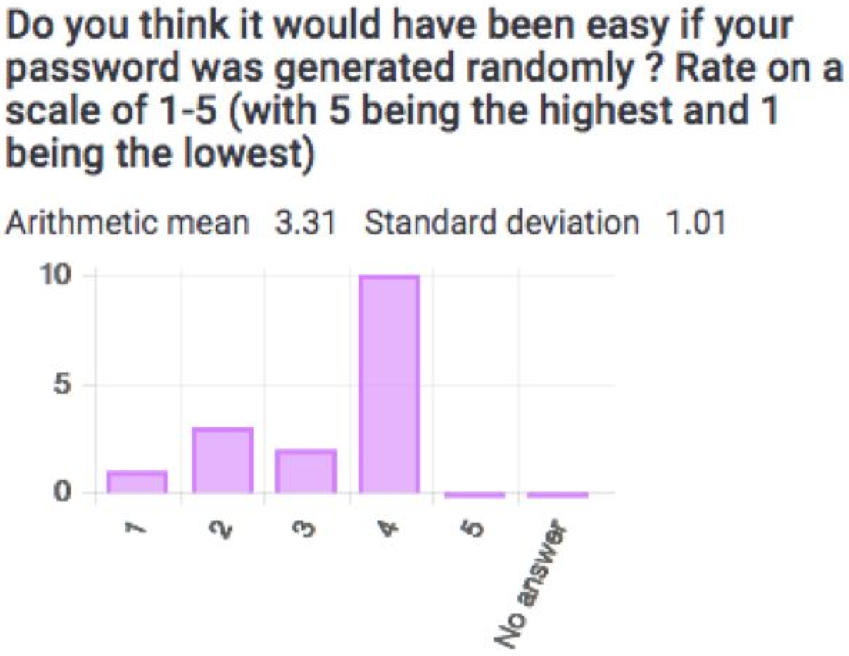
Arithmetic mean 3.5 Standard deviation 1.03

Question 7



Do you think if the password on the screen for a while, it helps the user memorize it quickly ?

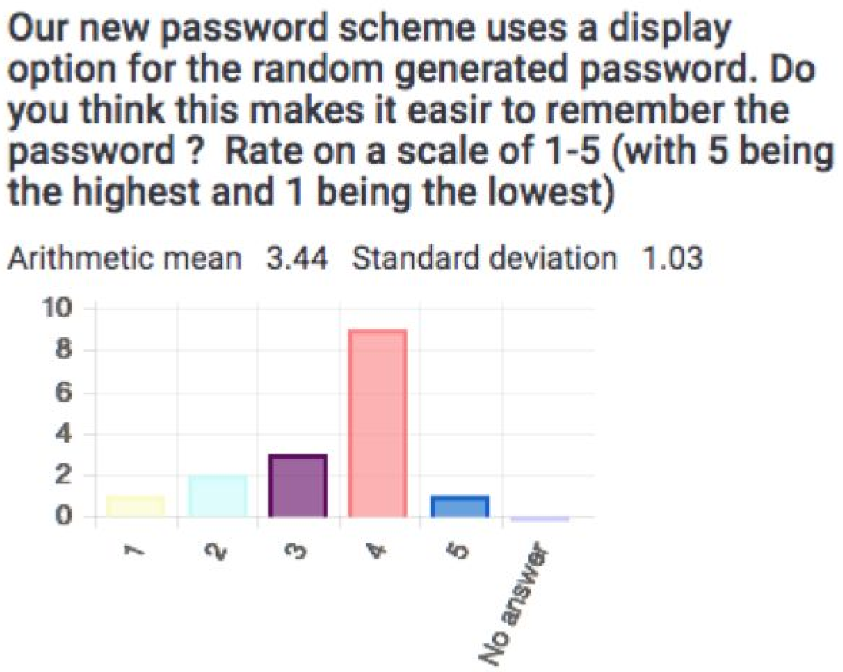
Question 8



Do you think it would have been easy if your password was generated randomly ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

Arithmetic mean 3.31 Standard deviation 1.01

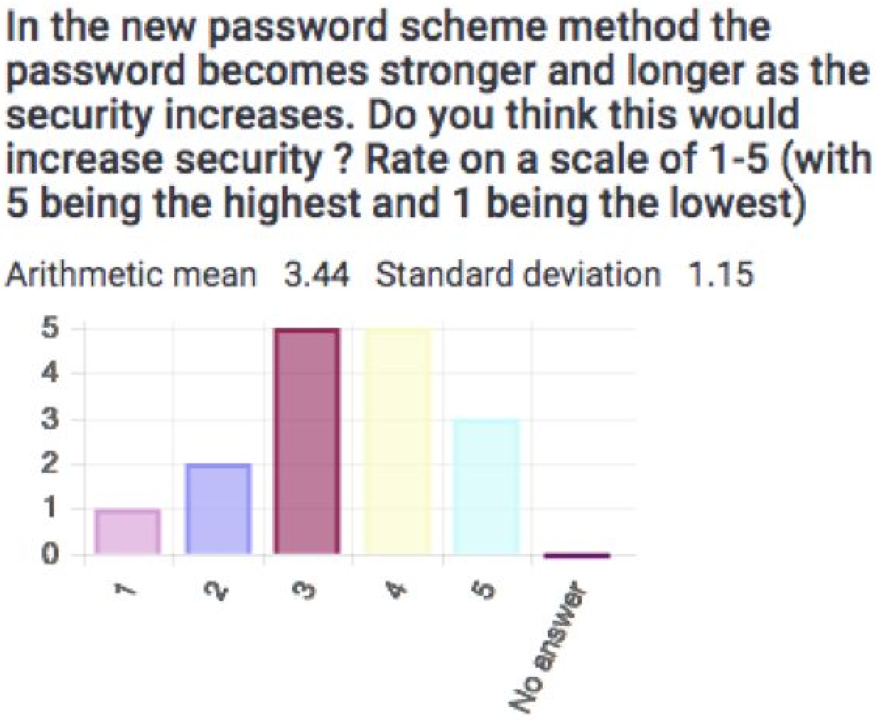
Question 9



Our new password scheme uses a display option for the random generated password. Do you think this makes it easir to remember the password ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

Arithmetic mean 3.44 Standard deviation 1.03

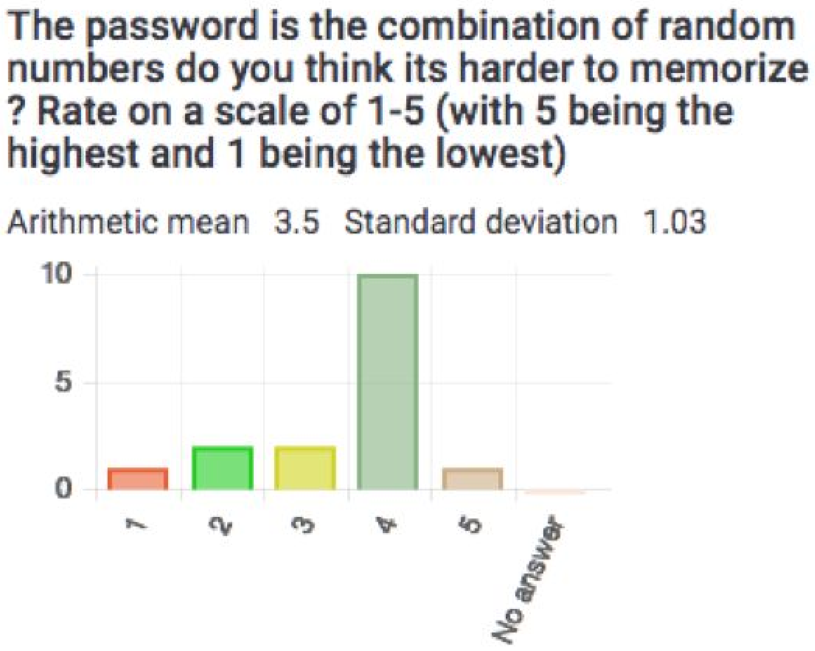
Question 10



In the new password scheme method the password becomes stronger and longer as the security increases. Do you think this would increase security ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

Arithmetic mean 3.44 Standard deviation 1.15

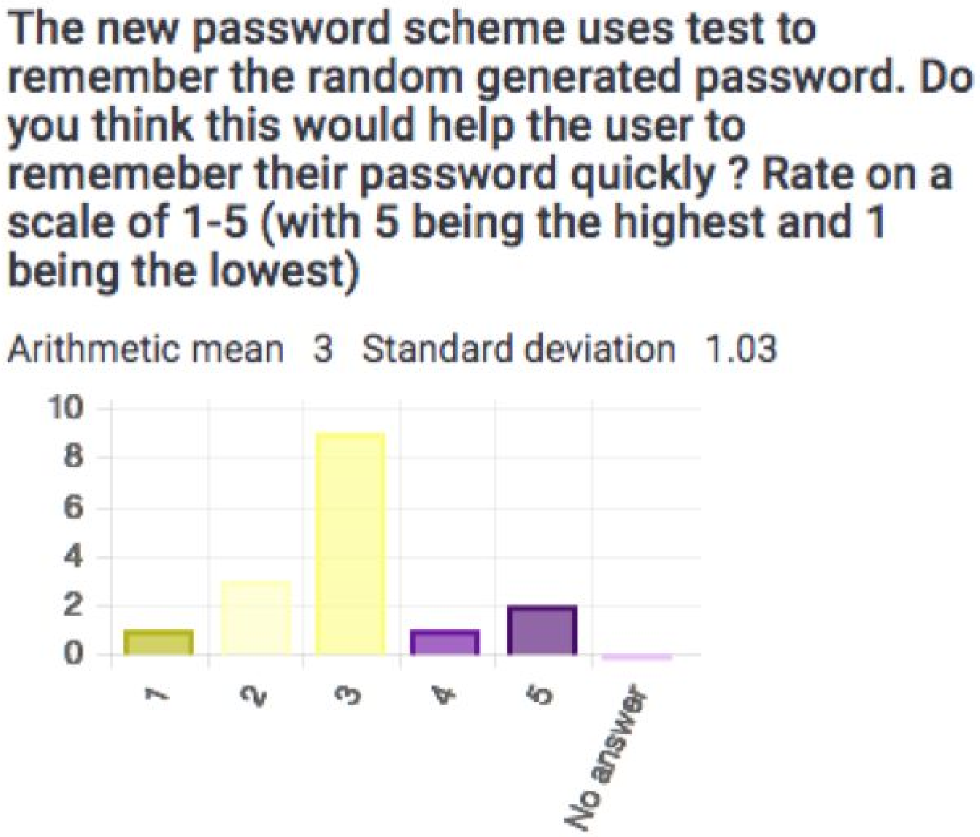
Question 11



The password is the combination of random numbers do you think its harder to memorize ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

Arithmetic mean 3.5 Standard deviation 1.03

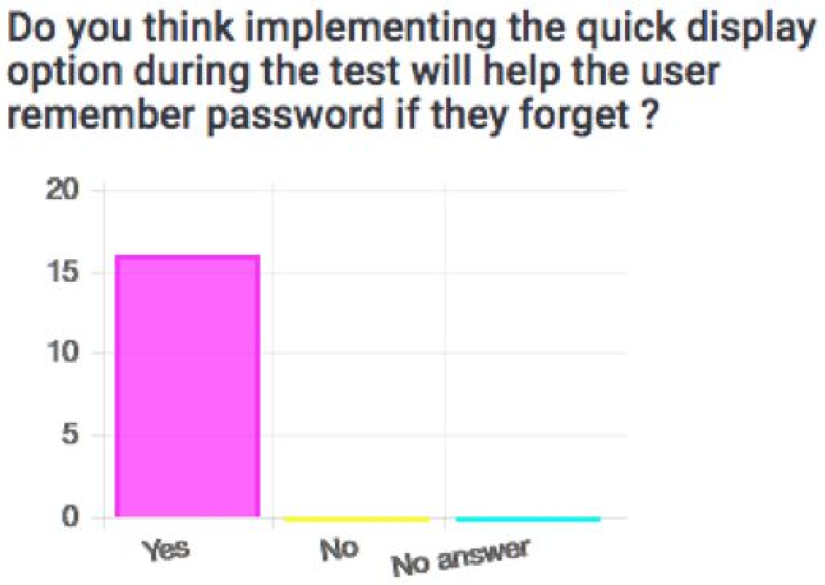
Question 12



The new password scheme uses test to remember the random generated password. Do you think this would help the user to rememeber their password quickly ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

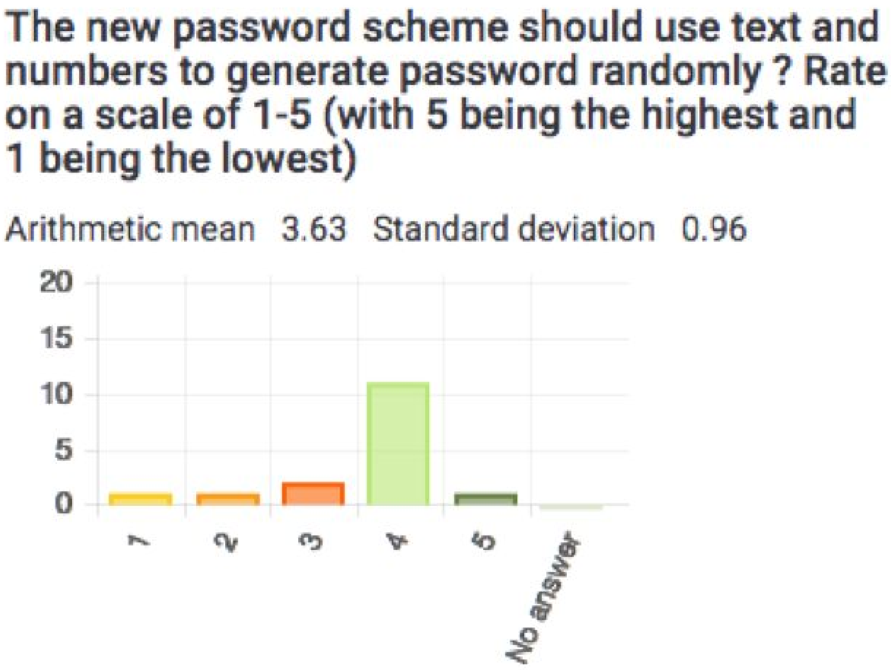
Arithmetic mean 3 Standard deviation 1.03

Question 13



Do you think implementing the quick display option during the test will help the user remember password if they forget ?

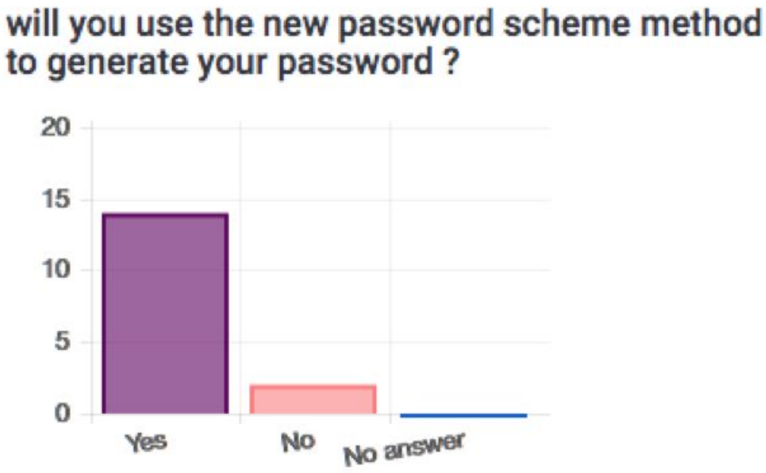
Question 14



The new password scheme should use text and numbers to generate password randomly ? Rate on a scale of 1-5 (with 5 being the highest and 1 being the lowest)

Arithmetic mean 3.63 Standard deviation 0.96

Question 15



will you use the new password scheme method to generate your password ?

2.5 Testing and Comparison

The results from the testing can be found in “stats.csv” in the “R” file. Below you will find organized tables of the CirclePass scheme results. They are identical to the tables for the schemes Text28, Imagept28 and Blankpt28 that can be found above in section 1.3. The source code that produces the tables and graphs can be found in “stats.R” in the “R” file. In the following section we will be comparing the Text28 scheme results to the CirclePass scheme results.

CirclePass
Number of logins per user

Username	Success	Failure	Total	Success%
12345	3	0	3	100.00
agent001	2	4	6	33.33
Bradley	2	3	5	40.00
cf	1	6	7	14.29
comp3008	3	0	3	100.00
janetseto	3	0	3	100.00
Jonny	3	0	3	100.00
Judah	2	5	7	28.57
mike	3	0	3	100.00
Mitch	3	0	3	100.00

nighthee	2	3	5	40.00
nobody	3	0	3	100.00
ocd12	1	6	7	14.29
qwer	2	3	5	40.00
Shane	2	5	7	28.57
SquatCobbler	3	0	3	100.00
Test	0	9	9	0.00
Tester	2	4	6	33.33
user5	3	0	3	100.00
username	2	3	5	40.00

Mean, median and standard deviation for logins

Logins	Mean	Median	Standard Deviation
Success	2.25	2	0.850696
Failure	2.55	3	2.723677
Total	4.8	5	1.908430

Mean Success% = 60.62

Successful login times per user

Username	Mean	Median	Standard Deviation
12345	7.939333	6.1790	3.3157443
agent001	3.169500	3.1695	2.6240733
Bradley	4.624500	4.6245	0.8959043
cf	14.332000	14.3320	NA
comp3008	6.036333	7.4320	2.9440541
janetseto	14.923333	15.7520	1.7044449
Jonny	5.995000	6.1950	0.5037152
Judah	4.585500	4.5855	1.8137289
mike	11.148333	10.0590	8.1011173
Mitch	10.432000	7.7900	5.7159222

nighthee	23.842500	23.8425	5.2969369
nobody	9.103333	7.4980	5.0040054
ocd12	49.356000	49.3560	NA
qwer	10.262500	10.2625	6.4608347
Shane	4.961500	4.9615	1.7444324
SquatCobbler	8.714333	9.1040	3.5117517
Tester	5.006500	5.0065	5.3902750
user5	19.680333	17.9060	5.1224092
username	6.354500	6.3545	3.8190837

Failed login times per user

Username	Mean	Median	Standard Deviation
agent001	9.151500	6.5790	7.597611
Bradley	7.601000	7.6740	4.796917
cf	7.476833	5.9845	3.752661
Judah	6.795200	7.4640	3.964832
nighthee	3.109000	0.9900	3.766753
ocd12	7.938000	2.6915	11.120705
qwer	8.985333	11.0470	7.614779
Shane	4.184000	5.1340	1.752267
Test	5.983889	4.6770	4.703367
Tester	5.020500	2.1630	6.799785
username	7.277667	2.6770	8.304240

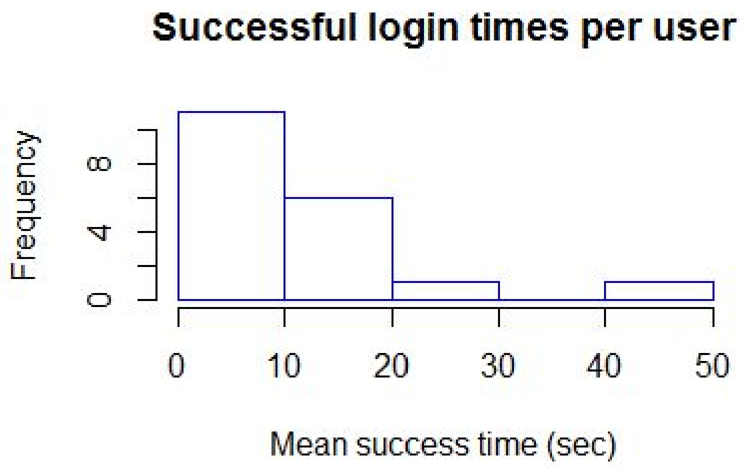


Figure 32 - Histogram for successful login times (CirclePass scheme)

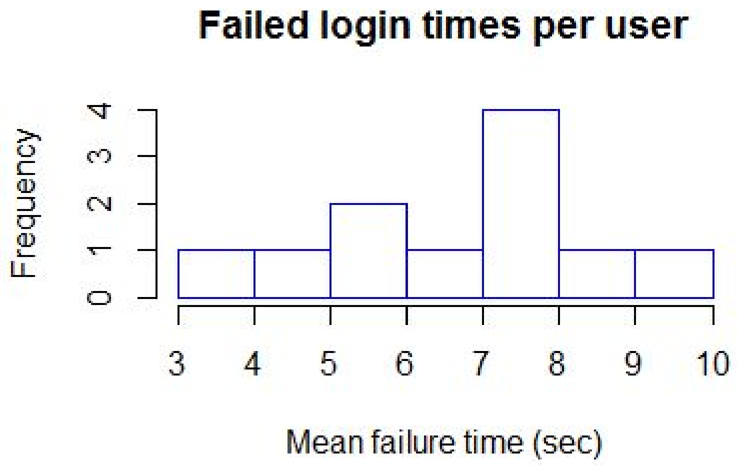


Figure 33 - Histogram for failed login times (CirclePass scheme)

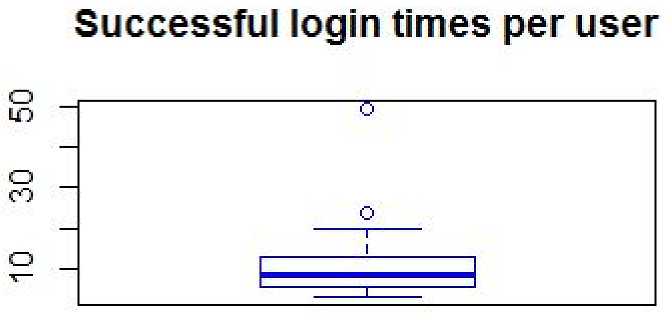


Figure 34 - Boxplot for successful login times (CirclePass scheme)

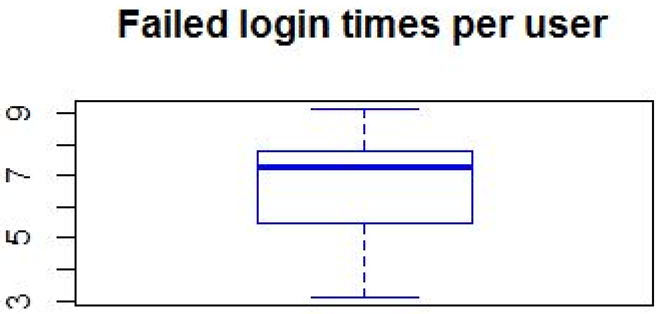


Figure 35 - Boxplot for failed login times (CirclePass scheme)

Interpretation of the CirclePass scheme statistics and graphs

In the above table *Number of logins per user* we can conclude the following:

- There were 20 users total.
- 9 out of 20 users got 100% success rate.
- One user failed all 9 attempts

In the above table *Mean, median and standard deviation for logins* we can conclude the following:

- The mean total login attempts is 4.8.
- The standard deviation for successful logins is 0.85 meaning the resulting number of successful logins are consistent between users.
- The standard deviation for failed logins is 2.72 meaning the resulting number of failed logins are not as consistent between failed logins.
- The standard deviation for total logins is 1.91 meaning the resulting number of total logins are consistent between users.
- The average user succeeded logins 60.62% of the time.

In the above table *Successful login times per user* we can conclude the following:

- Only one user dide not have a successful login.
- Half of the users had a standard deviation below 5.
- The highest standard deviation was 8.10.

In graphs that represent this table are *figure 32* and *figure 34*. From these graphs we can conclude that it takes the average user between 0-10 seconds to login successfully.

In the above table *Failed login times per user* we can conclude the following:

- Only 11 out of the 20 total users failed an attempt so the statistics are lacking.
- 6 out of the 11 users had a standard deviation below 5.

The graphs that represent this table are *figure 34* and *figure 35*. From these graphs we can conclude that it takes the average user between 7-8 seconds for failed logins. The mean failed login times were only between 3 and 10 seconds. We believe this is because when the user did not get the password the first time, some of them would press login quickly after failing to skip to the next password scheme resulting in low failed login times.

Comparing Text28 to CirclePass

To conclude we will be looking at the differences between the CirclePass scheme and Text28 scheme using the information above about the CirclePass scheme and the above information in section 1.3 about the Text28 scheme.

In the above section 1.3 we already concluded that Text28 scheme was the best of the three schemes for usability. The scheme has a success rate of 91.13% that could be increased if an outlier was removed. Also the time to login was an average of 5-10 seconds. When comparing that to CirclePass the same conclusion could be drawn. CirclePass has a success rate of 60.62%. This is the lowest success rate of the four schemes. This may be due to the way we tested. Users had 3 chances to guess the password. If they were correct it would count as 1 successful login. If the user failed, they were able to fail another 2 times for a total of three fails. So if they succeeded first try on the first test then failed three times on the second the result was a 25% success rate when it should have been 50%. Looking back we should have given the user three attempts and if they failed all three that would count as one fail. It may also be due to the password difficulty. All passwords were randomly generated. If the password was in a pattern that the user could easily remember they were more likely to succeed. We noticed that users would succeed on the larger CirclePass scheme (10x10) and the smallest password scheme (4x4). This may be because the smaller scheme was easier to remember, less boxes. And the larger one because it was the last scheme the user had to memorize.

The time taken to enter the CirclePass password was 0-10 seconds for successful logins and 7-8 seconds for failed login attempts. The successful login times are very good and can compete with the Text28 scheme of 5-10 seconds. There are a few factors that could affect the time. Some of our users tested on a touch screen computer, depending on the user this could be faster or slower than using a touchpad or mouse. Same goes with touch pads. If a user tested on the touch pad and had trouble using it it would result in a slower time.

To conclude, CirclePass can compete with the time to enter the password. Maybe if we tested with only touch screens the time may be lower but the success rate is where CirclePass falls. The 60.62% just does not compete with the 91% success rate of the Text28 scheme. Maybe with more testing we could find out how to increase the success rate. We already discussed what some problems may be but until CirclePass is updated, Text28 is the more usable password scheme.