COMP1531
T14A_AERO

Planning

Elicitation

A list of queries related to teamwork-driven communication Tools was asked to the list of interviewees listed below.

Questions :
1. How do you feel about the security of some of the teamwork-driven communication tools you've ever used?
2. How easy is it to access your account when you log in from some new device?
3. Have you ever forgotten the password to your account?
4. If you had a tool in your teamwork-driven communication tool that allows you to create a space where each member can update their progress, would you use it instead of communicating through normal chat?

Note that question 4 refers to the standups feature available in Treats.

Interviewees :
1. Hans (hanskr7882@gmail.com)
2. Rio (riowibowo8931@gmail.com)

Analysis & Specification - Use Cases

The user stores developed from the interviewees answers are as follows:
1. As a user, I want to be notified when some unknown attempts are made to log in to my account so that I will be aware when my account's security is at risk.
2. As a user, I want a 'did you forget your password?' option to appear when logging so that some leniency can be granted to me if I forget my password.
3. As a user, I want a space that allows all members in the channel to give updates about their progress so that the group can stay updated about the collective group's progress.

We also maintain some acceptance criteria for each user stories above :
User Story (1) :
1. After 5 failed attempts the user will be notified.
2. The user will be notified through email or a bot in the dms.
3. The user can reply to the notification to confirm that it is indeed the user and if 5 failed attempts are detected then the user has only forgotten his/her password.

User Story (2):
1. A small blue colored text 'did you forget your password?' added near the border where the user would input the user's password.
2. If the blue text is pressed the user's email will receive a message about the forgotten password incident.

3. Verifying the email allows the user to reset the user's password.

User Story (3):
1. Allow any user in the channel to start the space.
2. Notify all members except the member that started the space.
3. At the end of the space save the messages in the space where it is easy to locate.

Now, taking into account all the user stories and its acceptance criteria, we create use cases that solve each or some of the problems.
Use Case (Notify user of login attempts):
1. Notify the user via email if 5 failed login attempts are detected.
2. Provide a yes no question in the email to ask if it was the user that made the failed attempts.
3. If the user responds No, the user may also choose whether to block any log in attempts within some period of time determined by the user, the user would input the time in minutes.
4. Otherwise no other action is taken.

Use case (Forget password):
1. Blue text 'Did you forget your password?' appears near the password input box in the login page.
2. If the user clicks on it, wait for the user to enter the user's email.
3. If an email is entered, send a reset code to the email.
4. Wait for the user to put in the reset code and new password.
5. If the reset code and password is valid then the new password replaces the old password.

Use case (Standup)
1. Given some time length of the user's choosing, let the user start a standup in a channel that the user is part of.
2. All channel members receive a notification as if they've been tagged, except the user.
3. Pin the message on the channel at the end of the standup.

The use case for forget password is simply adding another case when the 'auth/passwordreset' path is used. On the other hand, the use case for standup optimizes the performance of the standups to better fit the user's needs. Notifying all members will increase the chance of more members participating in the standup, and

pinning the message will allow the group to inspect the information, such as each members' progress, accumulated at each standup.

Validation
The completed use cases above were then shown to the same interviewees to obtain their feedback.

	Notify login attempts	Forget password	Standup
Hans	This will ensure that I will be notified if any hacking attempts were to be made. It would be better if I can reach out to the app's help staff if anything like this were to happen.	This is a very useful addition, some sites/apps doesn't allows the user to immediately log in ,therefore the user may forget his/her password since the user will rarely have to remember it	Might be a useful addition, if standups are used to keep track of progress then i don't see why not include these features.
B	5 attempts are unnecessarily lot. 3 attempts are enough in my opinion because when a user forgets his/her/their password, he/she/they tend to just reset the password	This is a mandatory feature in my opinion, since nowadays there is plentiful websites and applications' password to be remembered; as a result, resetting password is not a rare occurrence.	Great feature to have, might as well develop more complimentary features.

Interface Design

For the forget password implementation, we can use the 'auth/passwordreset' path that is implemented in iteration 3 of Treats. The interface for this, displayed below, will then be the same as the interface provided in iteration 3.

auth/passwordreset/request/v1
 	
 Given an email address, if the email address belongs to a registered user, send them an email containing a secret password reset code. This code, when supplied to `auth/passwordreset/reset`, shows that the user trying to reset the password is the who got sent this email. No error should be raised when given an invalid email, as that would pose a security/privacy concern. When a user requests a password reset, they should be logged out of all current sessions.
	POST	**Body Parameters:**
		 `{ email }`
		
		 Return type if no error:
		 `{}`
			N/A

auth/passwordreset/reset/v1
 	
 Given a reset code for a user, set that user's new password to the password provided. Once a reset code has been used, it is then invalidated.
	POST	**Body Parameters:**
		 `{ resetCode, newPassword }`
		
		 Return type if no error:
		 `{}`
			400 Error when any of:
			 • resetCode is not a valid reset code
			 • password entered is less than 6 characters long

Figure 1.1 Password reset interface

The interface for notify user will be as follows:

auth/notify/failedattempts/v1 Given an email address, if the email is a registered user, send the user an email with a yes no query to confirm if the user is the one that did the failed login attempts. If the user verifies that the attempts were not made by the user, provide a link to the user that brings the user to the path below.	POST	Body Parameters: { email } Return type if no error: { }	N/A

Figure 1.2 Notify failed attempts interface

auth/login/blockv1 Given time, blocks any login attempt to this user's account within the time period given.	POST	Body Parameters: { time } Return type if no error: { }	400 Error when time is not valid.

Figure 1.3 Block login interface

And finally, the interface for standup will be mostly the same as the one provided in the iteration 3 interface but with some added cases.

standup/start/v1

For a given channel, start a standup period lasting `length` seconds.

During this standup period, if someone calls `standup/send` with a message, it will be buffered during the `length`-second window. Then, at the end of the standup, all buffered messages are packaged into one message, and this packaged message is sent to the channel from the user who started the standup: see section 6.13. for more details. If no standup messages are sent during the standup, no message should be sent at the end.

POST

Body Parameters:
`{ channelId, length }`

Return type if no error:
`{ timeFinish }`

400 Error when any of:
- channelId does not refer to a valid channel
- length is a negative integer
- an active standup is currently running in the channel

403 Error when:
- channelId is valid and the authorised user is not a member of the channel

Figure 1.4 Standup interface

Figure 1.4 shows the interface of standup provided in interaction 3 of treats, to fit our use case we add conditions that all members of the channel receive a notification of the standup. Furthermore, the standup message will be pinned to the channel at the end of the standup.

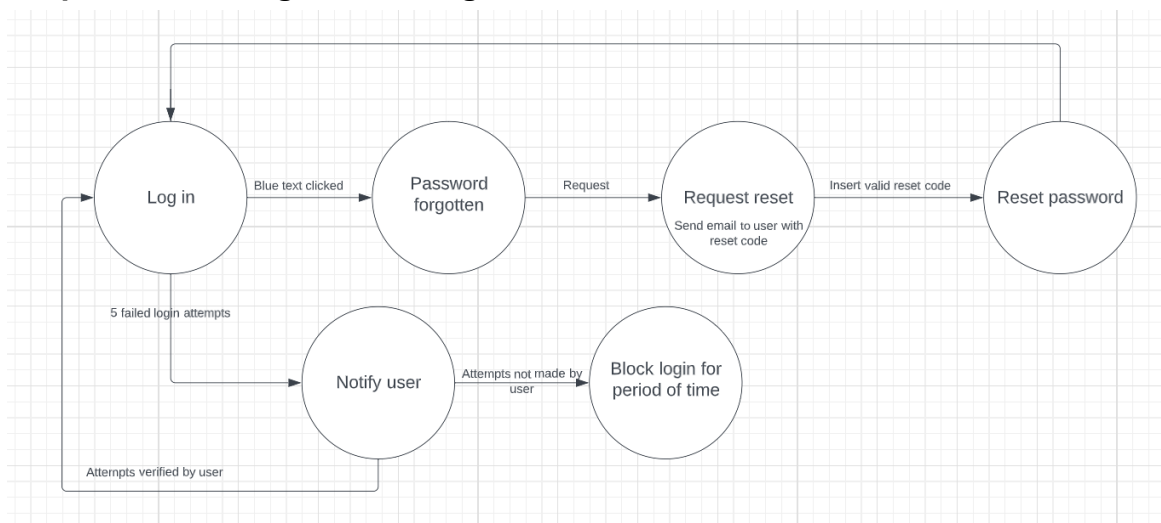## Conceptual Modelling - State Diagrams

Figure 2.1 Notify and password forget state diagram

Figure 2.2 Optimized StandUp diagram