

# **Bolt**

**Team 6 - Product Backlog:**

**Team**

**Nathan Peercy, Jia Lin Cheoh, Samuel Klarquist, Alex Willis, Siddharth Shah**

## **Problem Statement**

Private organisations and teams need secure and private platforms to share files and memos for internal transparency and to gather approval. Many traditional file sharing systems do not have features for memo or document approvals. In this space there is a need to unify the private scalable file sharing and document/memo actions in one platform. This app will allow private organisations and teams to securely share files, memos and seamlessly allow approval processing.

## **Background Information**

### **Problem**

Professional teams have the problem of secure and privatized file sharing and processing approvals. Team members have to go through a lot of trouble of getting physical copies of memos, documents etc. and then process approvals offline that need to be then uploaded to a digital pool. Existing services like Dropbox and Google Drive have the secure file sharing features embedded into them, but the issue of approvals in a team setting is not solved. Also, such platforms are not necessarily tailored for professional settings. There is no app out there that combines file and memo sharing for teams and approval processing. The target users are mainly professional teams looking for ways to fasten document and memo approvals and share them privately and securely.

### **Similar Applications**

There are platforms for file sharing like Dropbox, Box, Google Drive, OneDrive, Box, SugarSync, Tresorit and there are platforms for online document signing like DocuSign that allow document processing. Email is also used for sharing information via text.

### **Limitations**

Each of these apps have limitations when it comes to a combination of document/file/memo sharing and approval processing. DocuSign has digital signatures but not a single data pool for team files. Cloud storage apps have data pool features but no approval processing features. Email also has no built in approval processing functionality.

## Environment

Our frontend will mainly be written with react.js and deployed using node.js. We will be using Bootstrap as our main styling framework. We will be setting up webpack with SCSS for the styling of react. We will also be using Redux together with React if needed for more complex frontend features.

Our backend will mainly run Flask microframework and an IPFS node to upload files to the network. We will be using mySQL for account management and memo/file storage. For testing python code, we'll be using the pytest automated testing framework.

## Functional Requirements

Backlog id	Functional Requirements	Hours	Status
1	As a user, I would like to have an account	2	Sprint 1
2	As a user, I would like to securely login.	4	Sprint 1
3	As a user, I would like to be able to delete my account	2	Sprint 1
4	As a user, I would like to be able to login with two-factor authentication	2	Sprint 2
5	As a user, I would like to be able to reset my password using forgot password	4	Sprint 1
6	As a user, I would like to be able to update my account information when needed	4	Sprint 1
7	As a user, I would like to be able to send a memo	8	Sprint 1
8	As a user, I would like to send a memo to	4	Sprint 1

	multiple people		
9	As a user, I would like to be able attach a file to a memo	8	Sprint 2
10	As a user, I would like to reply to a memo	4	Sprint 1
11	As a user, I would like to indicate memo priority	4	Sprint 2
12	As a user, I would like to save drafts of my memos	2	Sprint 1
13	As a user, I would like to open drafts of my memos	2	Sprint 1
14	As a user, I would like to delete drafts of my memos	2	Sprint 2
15	As a user, I would like to have the ability to see the approval status of my memos.	4	Sprint 2
16	As a user, I would like to approve or reject a memo.	8	Sprint 2
17	As a user, I would like to see memos that I have sent.	3	Sprint 1
18	As a user, I would like to be able to have a list of memos sent to me.	3	Sprint 1
19	As a user, I would like see the dates and time my memos has been uploaded.	2	Sprint 2
20	As a user, I would like to delete memos I have received.	4	Sprint 2
21	As a user, I would like the UI to be optimized for mobile platforms	8	Sprint 2
22	As a developer, I would like my API server to be secure.	4	Sprint 2
23	As a user, I would like to set deadlines to memos that require approval processing.	4	Sprint 2
24	As a user, I would like to see the memos assigned to me for approval separately.	4	Sprint 2

25	As a user, I would like to be able to log out of an account	4	Sprint 1
----	---	---	----------

## Non-functional Requirements

### Usability

Bolt's UI should be easy to follow for a professional with no previous experience with the system. To further this goal, the UI should be minimalist with no unnecessary buttons or features cluttering the screen. We will have an intuitive user interface with a very short learning curve. The UI will also be easy on the eyes such that we will employ color schemes that are both professional, smooth and in sync with the functionalities. We will also ensure that our software complies with the 10 heuristics rules in user interface design.

### Architecture And Performance

The backend server is going to be written in Python and the frontend will be served with NodeJS. This will allow us to be able to scale the number of queries handled. The goal is to increase the ability of the server to handle heavy request loads.

We will enforce a consistent coding style. Code modifications will be merged via a pull request, which will be reviewed and approved prior to merging. Code reviews will check that coding standards are met and will look for introduced bugs.

### Security

The API server will have cross-origin domain checks that allow requests only from our frontend server URL and also secondary token checks with the OAuth service to authenticate proper requests.

## Use Cases

**Name:** 1 - Create an Account

**Case:** Creating an Account

**Steps:**

Action	System Response
1. Click on create account button	2. Window pops up to fill out username, password, and email.
3. User inputs information for 3 text boxes	
4. User clicks approve account button	4.returns to previous page

**Name:** 2 - Logging In

**Case:** Users log into their accounts

**Steps:**

Action	System Response
1. Click "login"	2. Send user OAuth
3.login	4.send user back to site
5. Display Home Screen	

**Name:** 3 - Delete account

**Case:** User deleting account

**Steps:**

Action	System Response
1. Go to profile Window.	
2. Click delete account	3. Dialog box pops up "Would you like to delete your account?"
4. Click "Yes"	4. Dialog box disappears

**Name:** 4 - Two Factor Authentication

**Case:** User authenticates using 2-factor auth

**Steps:**

Action	System Response
1. Click Login	2. Redirects to OAuth Login Page
3. Enter username/password	

4. Click Login	5. OAuth sends sms with code and redirects to 2-factor auth page
6. Enter the code	6. Callback redirects to the app home screen if successful or error if not.

**Name:** 5 - Forgot account password

**Case:** Users retrieve their password

**Steps:**

Action	System Response
1. Click on login screen	2. Display login dialog box, including a link titled "Forgot password?"
3. Click on the "Forgot password?" link	4. Send an reset email to the account holder's email
5. Click on the password reset link (redirects to special reset page)	6. Display dialog box for a new password
7. Enter new password and click save button	

**Name:** 6 - Update information

**Case:** A user wishes to change their account information

**Steps:**

Action	System Response
1. Click "update information" button	2. Display a screen with the user profile's information
3. Click on edit button next to the piece of information that is to be changed	4. Change the static box containing the information to an editable one
5. Make desired edits and click the save button	6. Save changes in the database

**Name:** 7 - Sending Memo

**Case:** Users want to send memos

**Steps:**

Action	System Response
1.Click on a “new memo” button	2. Open a memo editor
3. Edit title, text, and to content	
4. Click on attach file	5. Open file explorer
6. Select File	7. File explorer closes
8. Click Send	8. Memo Editor closes
	9. Toast shows with success/failure

**Name:** 8 - Memo sharing

**Case:** A user wishes to share a memo with multiple people

**Steps:**

Action	System Response
1.Select a memo and click the “share...” button	2. Open a textbox that allows the user to input the accounts of the users with whom the memo is to be shared
3. Enter the accounts of the target users	4. Add the memo to the “shared with me” list of the user with whom the memo is shared

**Name:** 9 - Attach a file

**Case:** Users want to upload their memos / files.

**Steps:**

Action	System Response
1.Click on the “upload” button.	2. Users choose the files to upload from their local machines.
3. Users click “open”.	4. Files uploaded and attached to selected memo

**Name:** Replying to a Memo

**Case:** Users want to reply to a memo sent to them.

**Steps:**

Action	System Response
1. Select a memo and click the “reply” button	2. Open a new Memo Editor that contains the original memo quoted.
3. Edit title and text content	
4. Click Send	5. Memo Editor closes
	6. Toast shows with success/failure

**Name:** 10 - Memo priority

**Case:** Users want to set the priority of their memos.

**Steps:**

Action	System Response
1. When editing a memo, click the “edit priority button”	2. Display a checkbox dialog with the options low, medium, and high
3. Click on the desired checkbox	4. Update the memo’s priority

**Name:** 11 - Save Memo Draft

**Case:** User wants to save a memo draft

**Steps:**

Action	System Response
1. Click on save draft button on create memo box	2. System saves all work on the memo, displays “Saved” toast

**Name:** Open Memo Draft



**Case:** 12 - User wants to open a memo draft

**Steps:**

Action	System Response
1.click memo from memolist	2. Displays memo editor, with fields populated with the selected memo

**Name:** 13 - Delete Memo Draft

**Case:** User wants to delete memo draft

**Steps:**

Action	System Response
1.click memo from memolist	2. Remove memo from system
3. "Memo deleted" toast displayed, close when ready	

**Name:** 14 - See approval status

**Case:** User wants to see approval status

**Steps:**

Action	System Response
1. Click on the desired memo.	2. "Approval" section indicates either "documents under review", "approved" or "rejected".
3. Close the memo display	4. Return to the list of memos

**Name:** 15 - Approve Memo

**Case:** A user who wants to look at the approval status of their files / memos

**Steps:**

Action	System Response
1. Click on the memo from the list of memos	2. Display memo details that contain the memo text, attached files and file details, and date sent and approve and reject buttons
3. Click the “Approve” button	4. Approve and reject buttons disappear, and the memo is marked as approved.

**Name:** 16 - See Memos shared with me

**Case:** Users want to see memos shared with them

**Steps:**

Action	System Response
1. Click on shared memos tab	2. Opens list of shared memos
3. Click on pertinent memo	4. Memo displayed

**Name:** 17 - See Memos I’ve shared

**Case:**

**Steps:**

Action	System Response
1. Click on shared memos tab	2. Opens list of shared memos
3. Click on pertinent memo	4. Memo displayed

**Name:** 18 - Seeing the date and time memos have been uploaded

**Case:** User wants to see the date and time memos have been uploaded

**Steps:**

Action	System Response
1. Click on the particular memo	2. The date and time memo has been

	uploaded will be displayed at the top of the memo.

**Name:** 19 - History of memos uploaded

**Steps:** See history of memos uploaded

**Case:**

Action	System Response
1.Click on the “Sent Memos” button.	2. A screen pops up with all past memos uploaded.
3.Hit the “close” button	4. Screen for history of memos closed.

**Name:** 20 - Deleting a received memo

**Steps:** Delete a memo

**Case:**

Action	System Response
1.Click on a “delete memo” button	2. Memo will be deleted from the user’s account.
3. Return to the original screen user was on.	

**Name:** 21 - UI Optimization

**Case:** Look at mobile screen

**Steps:**

Action	System Response
1.Navigate to website on mobile device	2. Displays login screen without any loss of functionality


**Name:** 22 - Secure Requests

**Case:** User wants to secure requests

**Steps:**

Action	System Response
1.Send post request with uid...	2.Verifies
	3. Does task or returns error

**Name:** 23 - Set deadlines

**Case:** User wants to set deadlines

**Steps:**

Action	System Response
1. In memo editor, click "Add Deadline"	2. Date input box opens
3. Enter future date	3. Memo indicates new deadline

**Name:** 24 - See Memos assigned to users for approval

**Case:** User wants to see memos assigned to them for approvals

**Steps:**

Action	System Response
1.Click on shared memos tab	2. Opens list of shared memos
3. Click on pertinent memo	4. Memo displayed

**Name:** Log out of an account

**Case:** Users wants to logout of an account

**Steps:**

Action	System response
--------	-----------------

1. Click “log out” button	2. Display “logged out successfully” message toast