

Project Requirements

dCloud: A Web3 Cloud Storage Mobile App

Prepared by: Simbad Marino

Tuesday, 5 April 2022

APP REQUIREMENTS

1. Scope

The purpose of this document is to list ALL functional and Non-functional requirements of dCloud app. This is considered a live document as requirements will change over time depending on end-user needs and feedback, industry trends, legal requirements, etc.

2. Functional Requirements

2.1 App Screens

2.1.1 dBrowser | File Explorer

Main Screen | Similar to box app¹ File manager screen which uses the BTFS API to:

2.1.1.1 Add individual files. (add,cp) | Prio0

2.1.1.2 Add empty folders (mkdir) | Prio0

2.1.1.4 Upload files to BTFS network (storage upload) | Prio0

2.1.1.5 Show upload progress ²(if possible using a modernized BitTorrent client sharding style animation by getting shard upload status) | Prio1

2.1.1.6 Rename, delete & move files and folders. (Make sure to rm CID to remove from repo) | Prio0

2.1.1.7 Filter, search and reorder files/folders (by name, date, etc) | Prio0

2.1.1.8 Use a modern look and feel similar to how box, google drive, etc manage files | Prio0

2.1.1.9 Storage cost + dev fee is displayed per each upload to BTFS network, a confirmation popup is shown to the user before submitting the upload. | Prio2

2.1.1.10 A banner is shown in the screen if the vault contract is not configured or balance is too low (WBTT Vault balance limits are configured in the Settings screen) | Prio3

2.1.1.11 Dev fee per uploaded file/folder is set to 3.3 BTT | Prio1

¹ See Picture 1 in References chapter

² See Picture 2 in References chapter

DCLOUD REQUIREMENTS

2.1.1 CID import option, this will import either a BTFS directory or file to current file system by pasting an already uploaded QmHash | Prio1

2.1.2 dBrowser | BTFS web explorer

This is an experimental website explorer which takes a BTNS key or BTFS CID to retrieve a complete website (using btfs get & cat) locally without needing a gateway service. (See [IPFS multi-page site](#) as reference)

2.1.2.1 This screen renders any provided BTFS CID or BTNS key as a normal website without needing an external gateway | Prio2

2.1.2.2 The screen have an upper “web browser like” search bar to input your CIDs or keys. | Prio2

2.1.2.3 There should be a way to bookmark all your favorite sites similar to how modern web browsers manage bookmarks | Prio3

2.1.2.4 Navigation arrows are included in the UI to “Show Previous page” or “Show Next Page” | Prio3

2.1.2.5 Recommended BTFS hosted sites are accessible trough a button at the right side of the search bar | Prio3

2.1.3 Wallet Manager

This screen is used by the final user to access his BTTC wallet and should be able to swap between the different tokens (BTT, WBTT, WBTT Vault). This section will be improved over time to keep it as simple as possible, final target is for the user to only interact with BTT and the rest should be managed in the background by the app backend (As an advanced feature, token swaps and each token balance must be accessible to users willing to deal with it)

2.1.3.1 Balance is visible for each token and updated every 10s when this screen is visible. Manual refresh is also possible by setting a RefreshControl. | Prio2

2.1.3.2 Tokens swap is available and UI is similar to popular token swap services (sun-swap, [JustMoney.io](#), pancake-swap, etc) | Prio2

2.1.3.3 General UI must adhere to common mobile crypto wallets look and feel (Tronlink, Metamask, TrustWallet, etc) | Prio2

2.1.3.4 Having a BTTC / BTT logo somewhere in the screen is a “nice to have” but not mandatory. | Prio4

2.1.3.5 Swap dev fee is set to 5.5 BTT per transaction. | Prio2

2.1.3.6 Transaction fee and dev fee is displayed with on a popup confirmation before performing any manual swap transaction. | Prio2

2.1.3.7 It must be possible to display a QR code with the public address upon user request ("Receive" button). | Prio2

2.1.3.8 An option to send out BTT or WBTT to any BTTC address is mandatory. | Prio3

2.1.4 Terminal Screen

A terminal screen is must likely a necessary inconvenience at this moment as go-btfs code and interface hasn't mature enough to make this optional (initial BTTC wallet setup, vault contract creation, btfs init & btfs daemon start are not accessible through API yet)

2.1.4.1 A terminal screen is mandatory at least until ALL essential BTFS features are linked to a UI component / feature | Prio0

2.1.4.2 If possible this terminal must be part "attached" to the ReactNative Screen and not a separate window /activity. | Prio2

2.1.4.3 Terminal emulator must have access to ALL BTFS commands. | Prio2

2.1.4.4 Android chosen terminal is currently termux, however if a more suitable and cross platform solution is identified developers are free to choose another option (specially if a shared golang library is feasible) | Prio1

2.1.5 Settings

A Settings screen is useful to configure the app features, define its behavior, Backup File System , private key + mnemonic, etc.

2.1.5.1 Define default storage duration. | Prio0

2.1.5.2 Monitor node status and BTFS version. | Prio0

2.1.5.3 BackUp file system option | Prio2

2.1.5.4 Backup BTTC Wallet PK and Node Mnemonic. | Prio2

2.1.5.5 Turn ON/OFF host and renter modes | Prio0

2.1.5.6 Define vault balance automatic filling thresholds. (In order to simplify and automate the BTT → WBTT → WBTT Vault swapping process to pay the storage network fees). | Prio3

2.1.5.7 Show total storage used and how much is locally available vs online available

2.1.5.8 Show available free storage until you have to pay for uploads.

2.2 Theme & Style

2.2.1 App theme must be dark by default. Light theme to be discussed later depending on demand. | Prio0

2.2.2 React Native styles and props must be declared on external .js files as much as possible to increase portability and code cleanliness. | Prio0

2.2.3 App colors must adhere to dCloud standard colors (currently: black, white, gray, and blue) whenever possible and must respect an homogeneous look. Changes are possible upon core team and dCloud graphic designers approval. | Prio1

2.2.4 Keep ALL visible text in a separate strings file to implement multi language support. | Prio0

2.3 App Life Cycle

2.3.1 When dCloud is used for the first time a welcome wizard is activated guiding the user on initial setup (will ask the user if they want to import an already existing Mnemonic or if they want to create a new one) | Prio1

2.3.2 dCloud stores in local non-volatile memory all user data including: | Prio0

- File System layout information (Complete files CIDs tree)
- Special app settings configured on Settings screen
- BTFS repository

2.3.3 dCloud WILL NEVER store (neither in NVM nor in RAM, including clipboard) any sensitive information such as Private Key or Mnemonic information. | Prio0

2.3.4 All UI screens must have the capability to auto refresh in a timely basis, the refresh rate should be short enough to be noticeable by users but not too fast so it reduces app performance or increases data consumption considerably. | Prio0

2.3.5 dCloud background service must consider an "Always ON" Internet traffic as mandatory to allow the BTFS daemon to run smoothly if Host mode is enabled. | Prio0

2.3.6 If dCloud is configured as “Renter Only” mode it must only keep the BTFS daemon on demand to reduce data usage and increase battery life. | Prio2

3. Non Functional Requirements

3.1 Architecture

dCloud must keep a modular architecture to reduce the code maintenance as much as possible, special and customized forks from projects like go-btfs are not recommended unless the development teams grows significantly and/or it is totally necessary to implement critical features.

3.1.1 dCloud modular approach must adhere as much as possible to the following: | Prio0

- go-btfs shared library / btfs binary (depending on execution approach)
- Cross Platform React Native UI
- Terminal emulator application (only if btfs compiled binary is used)

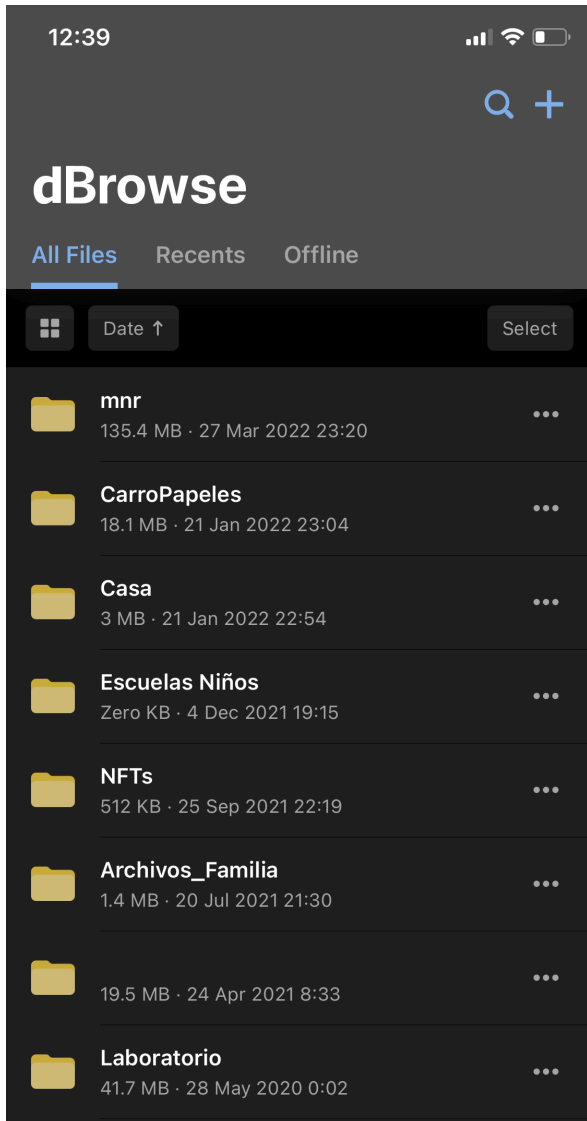
3.1.2 dCloud target market is to be supported by at least 80% of current Android(minSDK = 24) and iOS versions | Prio1

3.1.3 RAM and CPU consumption should stay in a safe level to avoid any sudden shutdown (specially on host mode). System exclusive / priority resources allocation is advisable whenever possible to keep the daemon running without problems. | Prio2

3.2 License and Legal

3.2.1 dCloud is Open Source and it uses several components from third party developers, before using any third party component, library, etc make sure there is no conflict with the use case dCloud is proposing.

REFERENCES



Picture 1: dBrowse screen reference



Picture 2 BitTorrent client sharding download progress
