|  |  |  |
| --- | --- | --- |
| *Use case name* | CreateAccount | |
| *Participating actors* | Initiated by *Client* | |
| *Flow of events* | 1. *Client* access the application by entering his name to login the system | |
|  |  | 1. If username doesn’t exist Server requests the *Client* to create an account by clicking the signup button |
|  | 1. *Client* Clicks the signup button and enters the client details and clicks the Create Account button | |
|  |  | 1. Server requests the *Client* to specify the path which will consider as the root path for the server. |
|  | 1. Client specifies the root path which is to be considered by the server | |
|  |  | 1. Server creates an account for the *Client* |
| *Entry condition* | * *Client should not have an account* | |
| *Exit conditions* | * New account is created for the *Client* | |
| *Quality*  *requirements* | * The Client account will be created in a minimal amount of time by the Server | |

|  |  |  |  |
| --- | --- | --- | --- |
| *Use case name* | ServerFailure | | |
| *Participating actors* | Initiated by *Server1* | | |
| *Flow of events* | 1. If *Server1* cannot process the request made by the Client, the control passes to the Server2 | | |
|  |  | | 1. Server2 automatically gets initiated and respond to the Client |
|  |  | 1. If Server1 and Server2 is failed then control passes to Server3 | |
|  |  | 1. Server3 automatically gets initiated and will respond to the client | |
| *Entry condition* | * *Server1* Should be failed * *Server1* and Server2should be failed | | |
| *Exit conditions* | * Server2 initiated automatically * Server3 initiated automatically | | |
| *Quality requirements* | * Connection will exist between the Client and server even if Server1 and server2 fails | | |

|  |  |  |
| --- | --- | --- |
| *Use case name* | FileShare | |
| *Participating actors* | Initiated by *Client1* | |
| *Flow of events* | 1. *Client1* uploads the file into the application and click the File Share button to share his file with Client2 | |
|  |  | 1. Server receives a request from the Client1 to share the file to Client2 |
|  |  | 1. Server process the request made by client1 and share the file to Client2 |
| *Entry condition* | * *Client 1 Should share a file* | |
| *Exit conditions* | * Server Share the file to Client2 | |
| *Quality requirements* | * Client2 Should receive the shared file without loss in data | |

|  |  |
| --- | --- |
| *Use case name* | Logout |
| *Participating actors* | Initiated by *Client* |
| *Flow of events* | 1. *Client* need to exit the application and clicks the log out button |
|  | 1. Server process the request and client gets   disconnected |
| *Entry condition* | * *Client* Should be logged in |
| *Exit conditions* | * The application is stopped and disconnected |
| *Quality requirements* | * Client successfully logged out of his application |

* 1. **Actors**
* Server1
* Server2
* Server3
* Client(Multiple Clients)
  1. **List of Use Cases**

Login

CreateAccount

Synchronization

Upload

Download

Delete

Modify

ServerFailure

FileShare

Logout

**Introduction:**

The purpose of the Drop box application is to provide a repository for the user to store the files and share the files with multiple users. This application will ease the process of users while documenting the same file. The user can upload, download, modify and delete the files using application. Shared files can be downloaded, modified and replaced the new version is updated in the repository.

**Scope:**

**Non functional equirements:**

**Usability**