RUP Artifacts

for

Pipe File Transfer (PFT)

Prepared by

- **Ahmed Nouralla**
- **Talgat Bektleuov**
- **Nikita Poryvaev**
- **Igor Mpore**

25/08/2021

Version History

Editor's Name	Date (DD/MM/YYYY)	Reason for Changes/Sections Updated	Version
Ahmed Nouralla	25/08/2021	Write functional requirements, user stories	1
Talgat Bektleuov	29/08/2021	Added non-functional requirements, constraints,	2
Igor Mpore	31/08/2021	Made a prototype of the project	3
Igor Mpore and Nikita Poryvaev	04/09/2021	Added some user stories and glossary	4

Table of Contents

Project repository on Github: https://github.com/sh3B0/pft

1. Business Goals and Objectives

- A web/desktop service for instant file sharing without the need to register and go through a long process is needed in the open-source community.
- The main goal is to allow simultaneous upload and download of the file
- Files should not be uploaded to the server. All files should go between clients only
- The service will provide a private and secure way to share files between two parties (or maybe more in the future).
- PFT should be free and easy to maintain and extend so that other features can be added

2. Roles and responsibilities of stakeholders

Stakeholder's Name	Roles	Responsibilities
Product Owner	Create concrete requirements	To manage the process and team and assign responsibilities.
Developer	Design and implement the application	Use technology and tools to produce the front-end and backend of the application.
User	Use and test the application	Send the feedback if he finds any bugs in the application

3. Requirement Analysis and Specifications

3.1. Features

ID #	User Story Title	Priority	Any Other Label
1.1	Generating a shareable link	Must	
2.1	Establish the connection between users using that link	Must	
2.2	Uploading the file(s)	Must	
2.3	Downloading the file(s)	Must	
3.1	Develop the website API.	Must	
4.1	Able to share generated links with telegram/WhatsApp/etc. friends using a button on the page	Good to have	

3.2. User Stories

Stakeholders	User Story Title	User stories
Web User	Link generation	As a user, I want to have a shareable private link to my files so that I can share them with my friends.
	File Upload	As a user, I want to upload my file to a data channel so that my friend can download it immediately while I'm still uploading.
	File Download	As a user, I want to be able to download files simultaneously with my friend so that I will be able to receive them quickly
	Front-end design	As a user, I want to have a nice user interface for the website so that I can use the service functionality easily.
	Sharing the link on social networks	As a user, I want to be able to choose a friend and send him a link so that I will not need to copy the link to Telegram/Whatsapp/etc.

3.3. Non-functional requirements

Category	Quality attribute	Explanation	How will we achieve it?
Usability	User interface aesthetics	Application has to have an easy-to-use interface for the user to be able to use it quickly and effectively	We will poll users for their opinion (1-5 stars) regarding the interface (ease of use, attractiveness) and collect statistics.
Security	Confidentiality	Files uploaded by users shouldn't be accessible from other users	Data channel will be encrypted, connection to the website and API will use SSL, one-time room ids.
Compatibility	Interoperability	The application should allow exchanging data (files) between different browsers	Tests will be developed and run on multiple browsers and/or operating systems.
Functional Suitability	Functional Correctness	Transferred files should arrive on the other side intact and without corruption.	If a new ICE candidate arrives, we use the existing channel to signal it, so it will be used later in case of session interruption.
Maintainability	Reusability	The application can later be embedded in another bigger system.	We will write the code in a modular paradigm.

3.4. Constraints

Constraint	Explanation
The browser should support WebRTC	The application relies on WebRTC protocol which should be supported by the user's browser.
Users can not be behind a symmetric NAT.	Symmetric NAT won't allow users to use STUN server to discover their public IP address for P2P communication.
Firewall rules have to be adjusted to allow P2P connection used by WebRTC	The firewall may intervene and block WebRTC requests, resulting in connection establishment failure.
Browsers extensions should not add JS code to the page that intervenes with requests.	Some browser extensions like VPN services might modify outgoing/incoming requests, resulting in upload/download failure/interruption.

4. Software Development plan

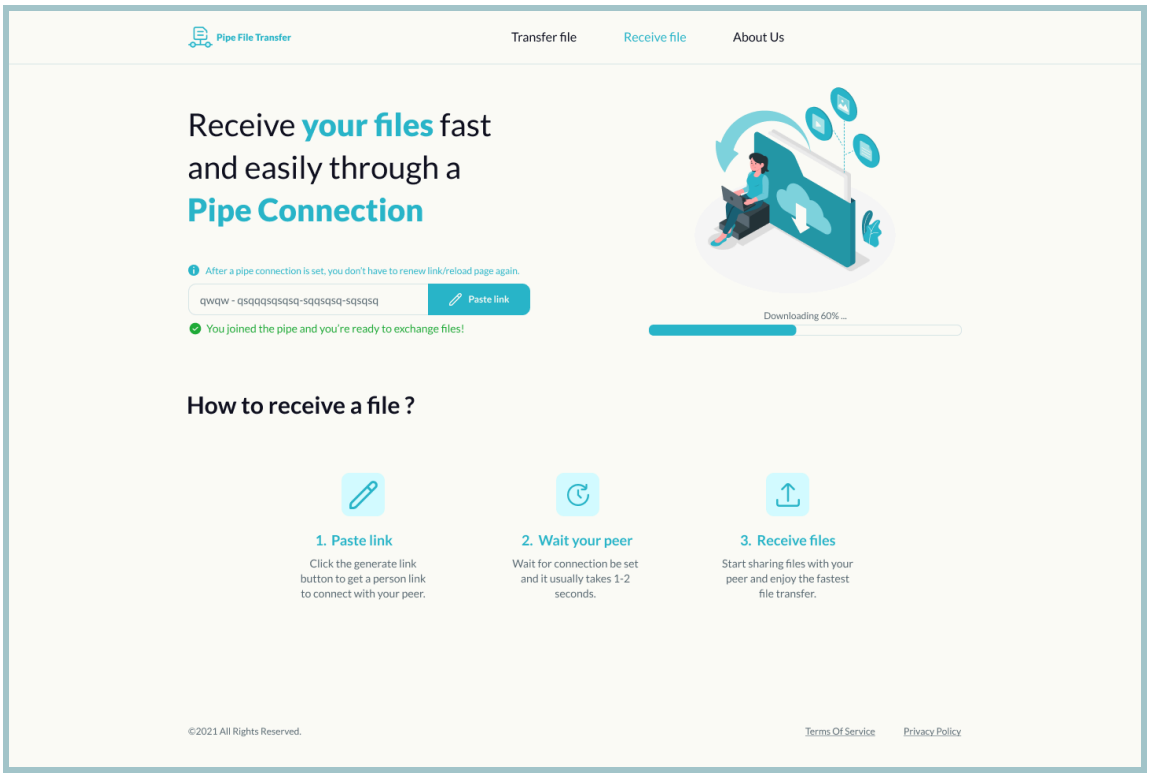
Inception Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	24/08/2021 - 30/08/2021	Product owners (Talgat Bektleuov, Ahmed Nouralla)	Determine Business goals and objectives with valid justification Identify the stakeholders Establish roles and responsibilities	Deliver the documentation of achieved milestones
#2	30/08/2021 -31/08/2021	Product owner (Talgat Bektleuov)	Requirement engineering (20% user stories) Identify Risks	Update the documentation of achieved milestones with User stories and Risk Lists

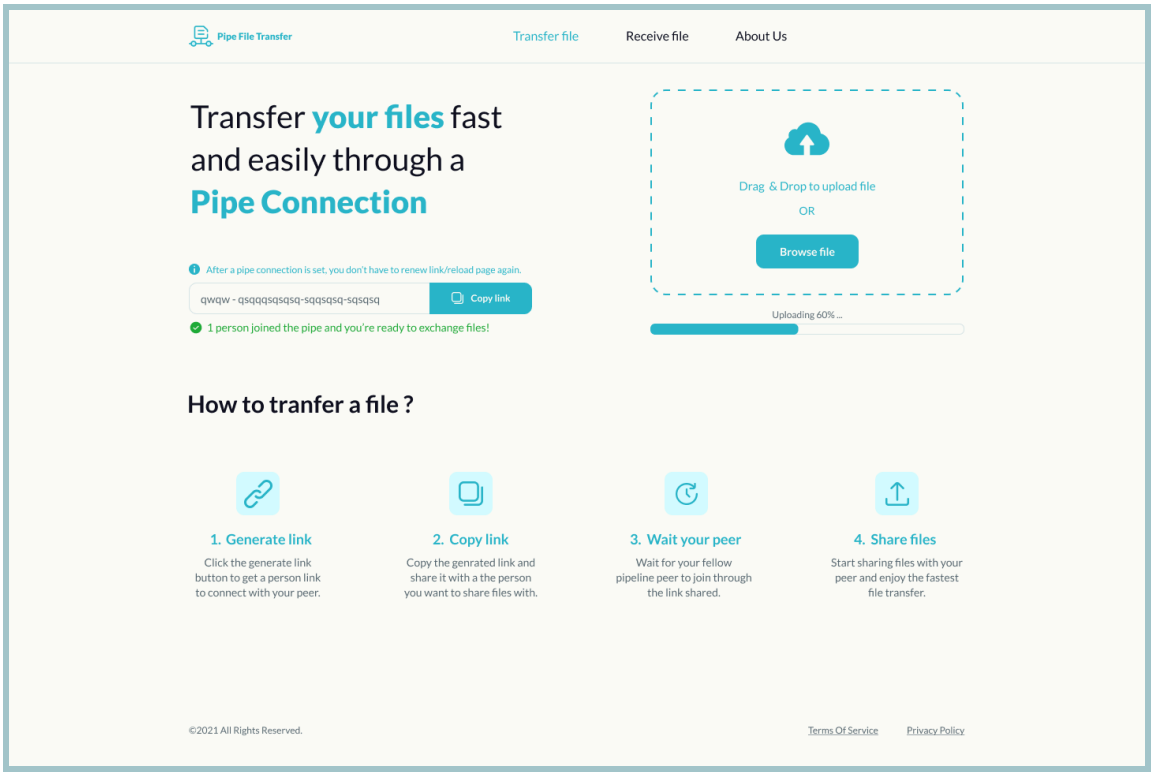
Elaboration Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	30/08/2021-2/09/2021	Ahmed Nouralla Talgat Bektleuov Nikita Poryvaev Igor Mpore	Revise User Stories (100%)	Document 100% of user stories
#2	6/09/2021-8/09/2021	Ahmed Nouralla Talgat Bektleuov Nikita Poryvaev Igor Mpore	Software development planning	Iteration Plan
#3	10/09/2021-15/09/2021	Ahmed Nouralla Talgat Bektleuov Nikita Poryvaev Igor Mpore	Software Architecture Test Plan	Software architecture document Test Plan Document

Construction Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	1/09/2021-10/09/2021	Ahmed Nouralla	Implement Feature 1.1, 2.1, user story 1 Unit test cases for features 1.1, 2.1	Working feature 1.1, 2.1 branches Unit test results
#2	11/09/2021-21/09/2021	Ahmed Nouralla	Implement Feature 2.2, 2.3, user story 2, 3 Unit test cases for features 2.2, 2.3	Working feature 2.2, 2.3 branches Unit test results
#3	06/09/2021 - 20/09/2021	Igor Mpore	Implement Feature 3.1, user story 4 Unit test cases for feature 3.1	Working feature 3.1 branch Unit test results
#4		Ahmed Nouralla, Igor Mpore	Implement Feature 4.1, user story 5 Unit test cases for feature 4.1	Working feature 4.1 branch Unit test results

Transition Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	20/09/2021-04/10/2021	Ahmed Nouralla Talgat Bektleuov Nikita Poryvaev Igor Mpore	Integration, End to end testing Training for Users and Developers	Github repository Merged branches Integration and ended to end test results
#2		Ahmed Nouralla Talgat Bektleuov Nikita, Igor Mpore	Final product release	Working Product Final README for developers and Users

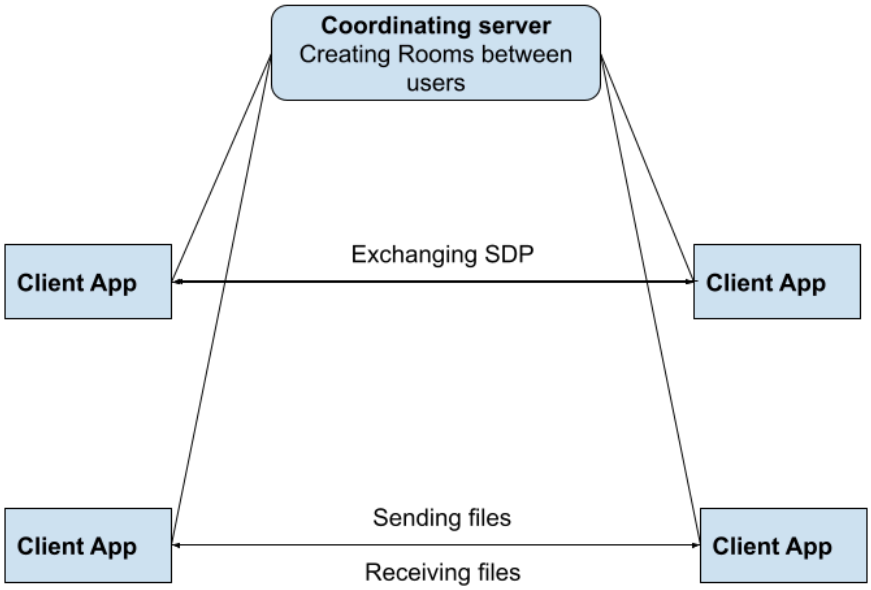
5. Prototype of the project (UI design)



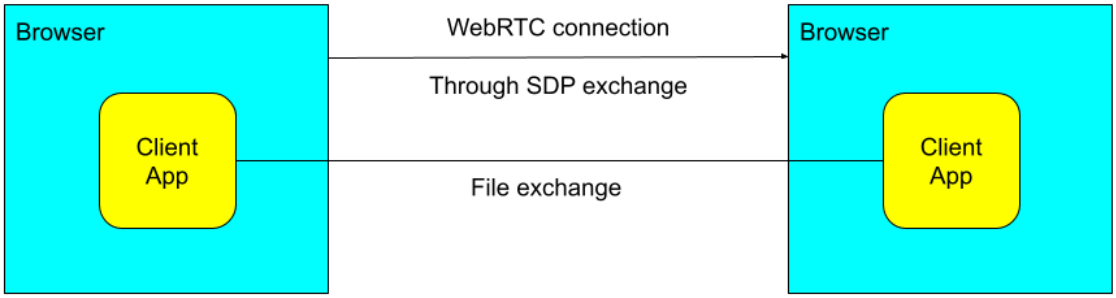


6. Architectural views

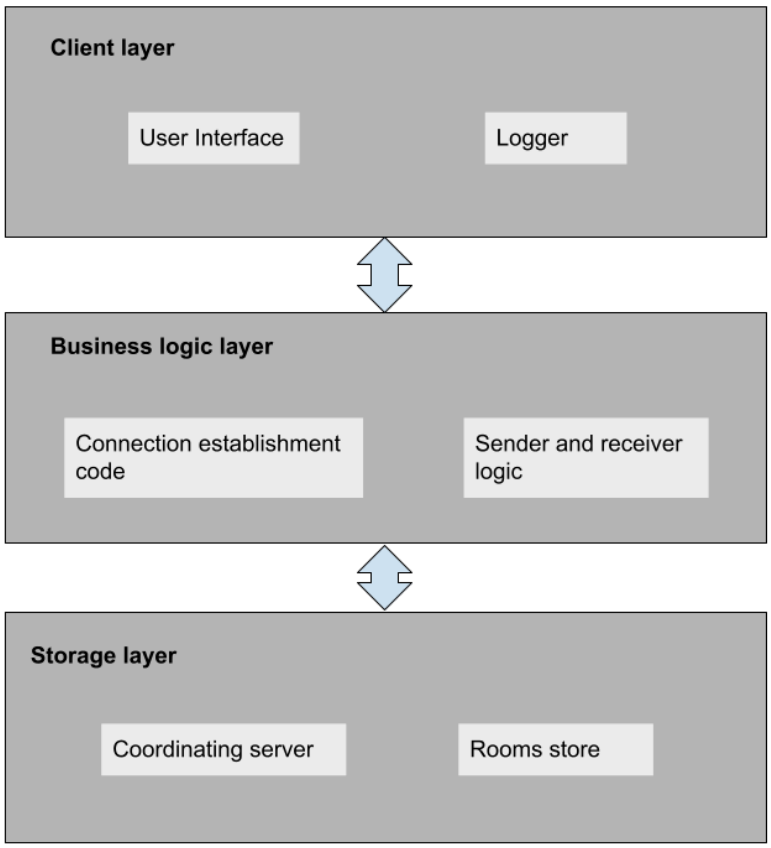
- Dynamic View



- Allocation View



- Static View



7. Project glossary

Term	Explanation
WebRTC	Set of protocols used Real-Time Communication over the web, used by our application to send files.
Data channel	The communication link between two users of the system
File transfer	Sending files from one user to another
Pipe	Data channel used by our system to send and receive files simultaneously.
SDP	Session Description Protocol, a string containing connection parameters used for establishing connection between sender and receiver.
Offer	Request containing SDP sent from file sender to receiver.

Answer	Response containing SDP from file receiver to sender.
Signaling	Exchanging SDP between sender and receiver through Offer/Answer requests.
Room	JSON file (with an id) on server storing request type (Offer/Answer) and SDP