

Extended BP

Corporate Task Completion Harvester
COMPLETE PRODUCT DESCRIPTION
A web service which allows users to establish tasks with specific deliverables which they expect from other people, colleagues, or counterparts. The task contains a set of subtasks, a deadline, and any information which might be necessary for the delivery of the required materials. Once the counterparty accepts the task, the web service sends periodic notifications and reminder emails with the aim of collecting the required deliverables. Deliverables could be anything (to be asserted as complete by Reciever and to be confirmed as complete by Sender) or documents (also can be uploaded). Tasks can contain subtasks which pertain to both parties in which case both parties receive updates with what they need to contribute. Normally, the assigning party receives updates and the delivering party receives reminders. Once tasks are marked as complete and both parties accept the completed state, notifications stop. The web service provides an unflashy report of how the work progressed (proof of work). This idea removes the need for constantly manually sending reminder emails and allows employees to declutter by centralizing pending tasks to a common workplace. It also allows people to easily create a document paper trail to later show as proof of potential negligence. Potential users include company employees in procurement and sales (or interdepartmental communication), students with homework or group projects, and general usage from activities which require contracts such as realstate/renting and others.
PROTOTYPE DESCRIPTION - REDUCED SCOPE
A web service which allows users to establish tasks with specific deliverables which they expect from other people, colleagues, or counterparts. Every task has a subtask (thing to be completed), a deadline, and any information which might be necessary for the delivery of the required materials (described below). Once a task has been given and accepted, the web service sends periodic reminder emails with the aim of collecting the required deliverables - in effect reminding the other party to deliver it. A single task has a single subtask to be completed. The party which creates the task is called the Sender, and the party which is expected to accept and complete it is called the Reciever. Once a task is marked as complete by the Reciever, should it also be accepted as such by the Sender, notifications stop and the task enters archival mode. Information about the task history can be seen on the task page. The service is entirely based on unique URL identifies (elaborated on below) which allows users to use it fully without the need to create an account. This idea removes the need for constantly manually sending reminder emails and allows employees to declutter by centralizing pending tasks to a common workplace. Potential users include company employees in procurement and sales (or interdepartmental communication), students with homework or group projects, and general usage from activities which require contracts such as realstate/renting and others.
THE X OF Y - THE WETRANSFER OF COLLABORATIVE WORK COLLECTION
WeTransfer does one thing and it does it really well - file transfers. It is simple, beautiful and dependable. It handles all necessary communication for users and it doesn't require unnecessary actions. It promotes artists and gets the job done. This is our goal. As simple as possible, with as few steps as possible.

Design Doc. I

Goals	Non-Goals
Build the idea as described in the idea description panel in the Extended BP above. The service will provide centralized deliverables collection services. It will have a minimalized interface and require very little to interface with it. It will send reminder and notification emails to two parties regarding the state of the task.	This version will not include any premium features, account creation and customization, file hosting, or billing options. Detailed functionality assignment for the prototype and full version can be seen below.

Functionality (SCOPE)					
PROTOTYPE	ACCESS	Can access full functionality with email	FULL	ACCESS	Plus version requires an account
	TASKS	Unlimited, accessed through email		TASKS	Unlimited, accessed through profile
	SUBTASKS	1 per task		SUBTASKS	Unlimited
	CONTENT	Note		CONTENT	Note, title, details, files, etc
	DEADLINE	1 per task		DEADLINE	1 per task, 1 per subtask
	COLLAB	1 other person		COLLAB	Multiple colab, self-tasks, colab work
	NOTIF. PERIOD	Automatic		NOTIF. PERIOD	Automatic + manual
	DELIVERABLES	Work is noted		DELIVERABLES	Some work can be uploaded (File Hosting)
	HISTORY	Notifications, actions, notes		HISTORY	Notifications, actions, notes, DW, public
	CONFIRMATION	Required		CONFIRMATION	Required
	APPROVE SUBM	Submissions must be approved by both		APPROVE SUBM	Submissions must be approved by both
	NOTIFICATIONS	Every single action		NOTIFICATIONS	Every single action
	CLONE TASK	None		CLONE TASK	Yes
	BILLING	None		BILLING	Mechanism integrated with registration
PASSWORD	None	PASSWORD	Optional		
E.REPLY = NOTE	No	E.REPLY = NOTE	Yes, email replies become notes.		
PROFILE	None	PROFILE	Can be accessed through account or link.		

Creating a Task (More in UI Sect.)	
TITLE	String, 50 chars
NOTE	String, 500 chars
RECIPIENT	Email
SUBTASK	1 subtask to be completed
DEADLINE	Datetime
NOTIF. FREQ.	Notification frequency

Subtask (checkbox)	
TITLE	String, 50 chars

Task history (chronological)	
NOTES	The history includes all notes
ACTIONS	The history includes all actions
NOTIFICATIONS	The history includes all sent notifications

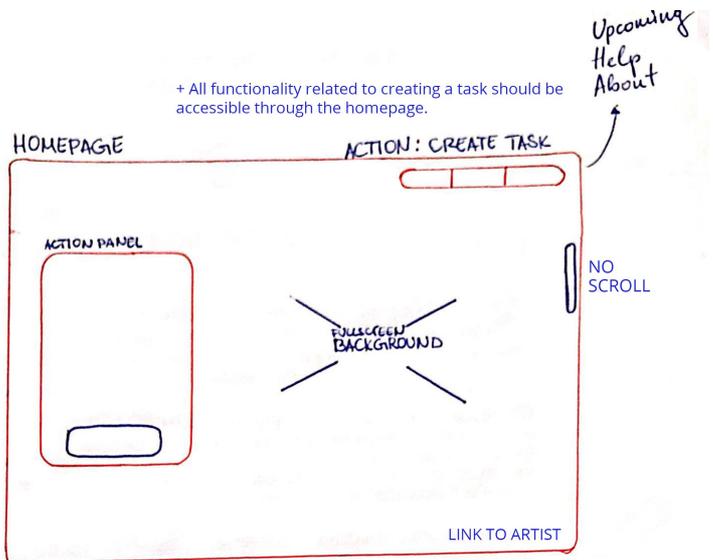
Functionality (PROCESS)		
1. SENDER	Creates a task and sends it to recipient. Anyone can create an unlimited number of tasks with their email.	
2. WEBAPP	Sends an email to recipient to confirm they accept the task	
3. RECIPIENT	Accepts or declines the task via email.	
	DECLINE	The app notifies the sender. The task countdown does not start. The sender can resend the task confirmation from Task page.
	ACCEPT	MOVE TO STEP 4.
4. SENDER	The sender can close the task at any time and optionally leave a note with reasoning. When closed, both parties are notified & the task is archived.	
5. RECIPIER	The reciever can abandon the task at any time and optionally leave a note with reasoning. Both parties are notified & the task is archived.	
6. WEBAPP	The app starts the task countdown and enters the notification/reminder loop. It notifies the sender.	
	LOOP	Is the task incomplete & w/in deadline? If yes, notify the receiver (notification frequency settings). Notify sender of any actions
	NOTES	The two parties can communicate via notes left on the task's page. The notes become part of the task history.
7. RECIPIENT	The recipient completes the task. They mark the task as complete in the App, specify their delivery method, and leave a note. The sender is notified.	
8. SENDER	The sender reviews the confirmation by the recipient and checks whether they have received the work.	
	DECLINE	The sender declines the work and leaves a note. The loop continues. The recipient is notified.
	ACCEPT	The task is marked as complete. Both parties are notified. The loop ends. The task page reflects this and shows the task history.
9. WEBAPP	The task page is archived and the history is accessible to anyone with a link. Once closed, a task cannot be reopened, but it can be cloned (full ver).	
10. WEBAPP	When the deadline is reached, Sender can extend the deadline. Task remains active indefinitely until closed, abandoned, or completed.	

Key Elements (ELABORATION)	
EMAIL ACCESS	Full functionality can be accessed by visiting a unique sender/recipient link in any of the received emails. Users with Sender/Receiver links can execute different actions. There is a separate link for public viewing. 3 links in total.
URLS	There are unique Sender/Receiver URLs. Based on which URL a page is reached by, different options are available. Things can be appended to them to signify different actions (Accept/Reject/Close/Abandon are just parameters appended to the Sender/Receiver links)
UNLIMITED TASKS	A new task is simply a matter of another set of links. This will not become too much hassle for users to maintain before the prototype stage is over. After that, we can have a profile link which displays all tasks (as sender/recipient) with admin functions & provides settings for notifications etc.
DEADLINES	When a deadline is reached, the task isn't paused or suspended, but users are notified. Task remains active indefinitely until closed, abandoned, or completed. Deadline text in emails has to reflect finished deadline. Notifications still continue periodically.
SCALABILITY	The architecture and scheduling algorithm should provide for potential scaling in the future.
NOTIFICATIONS	Notification frequency can be edited at any time from the Sender link. Options can include every "n" days, daily, weekly, hourly. There is a

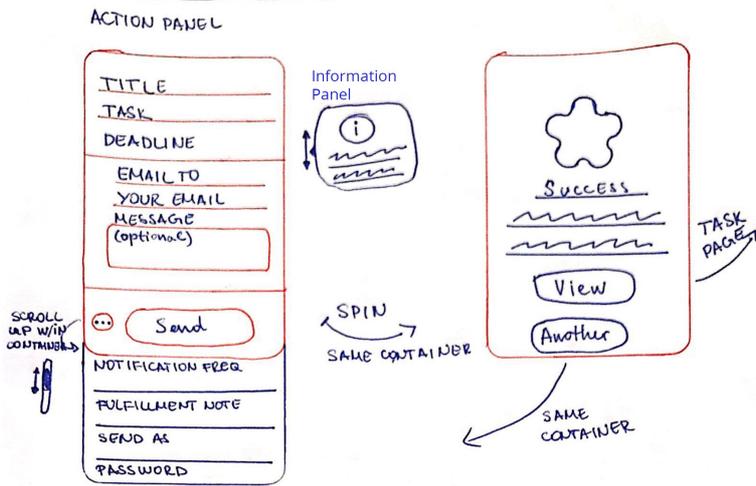
NOTIFICATIONS	scheduler running which schedules notification emails.
----------------------	--

EMAIL Communication			
TASK CREATED	SENDER	Confirmation for created task	Sender link. Confirmation shown on webpage for Sender.
TASK CREATED	RECEIVER	Invitation to accept the task	Link to confirm + edirect to receiver link.
TASK DECLINED	SENDER	Notification for declined task by receiver	Confirmation shown on webpage for Receiver.
TASK ACCEPTED	SENDER	Notification for accepted task by receiver	Sender link.
TASK ACCEPTED	RECEIVER	Confirmation for accepted task	Receiver link.
NEW NOTE	NOTE SENDER	-	Confirmation for sent note shown on screen & reflected in history.
NEW NOTE	NOTE RECEIVER	Notification for received note.	Link to note on task page appended to Sender/Receiver link.
PERIOD NOTIFY	RECEIVER	Notification for impending deadline and incomplete task status	Receiver link. Remaining time until deadline. Task details.
PERIOD NOTIFY	SENDER	Notification for sent reminder to Receiver.	Sender link. Remaining time until deadline. Total reminders sent.
NEW ACTION	RECEIVER	Confirmation for action	Receiver link.
NEW ACTION	SENDER	Notification for new action on task	Action details (title + delivery method). Link to accept/decline appended to sender link. Send note via Sender link.
ACT. REJECTED	SENDER	Notification for action rejected	Sender link. Note which comes with rejection.
ACT. ACCEPTED	RECEIVER	Notification for successful task completion.	Task statistics. Public link to history. Link to create new task.
ACT. ACCEPTED	SENDER	Confirmation for successful task completion.	Task statistics. Public link to history. Link to create new task.
TASK CLOSED	RECEIVER	Notification for task closed by Sender	Note for closing (if at all), Task statistics. Public link to history. Link to create new task.
TASK CLOSED	SENDER	Confirmation for task closed by Sender	Note for closing (if at all), Task statistics. Public link to history. Link to create new task.
ABANDONED	RECEIVER	Confirmation for task abandoned by Receiver	Note for abandoning (if at all), Task statistics. Public link to history. Link to create new task.
ABANDONED	SENDER	Notification for task abandoned by Receiver	Note for abandoning (if at all), Task statistics. Public link to history. Link to create new task.
DEADLINE HERE	RECEIVER	Notification that the deadline has been reached	Receiver link. Task statistics.
DEADLINE HERE	SENDER	Notification that the deadline has been reached	Link for extending the deadline appended to Sender link. Task statistics.
DEADLINE EXT.	RECEIVER	Notification for deadline extension	Note for extending the deadline (if at all), Task statistics. Receiver link.
DEADLINE EXT.	SENDER	Confirmation for deadline extension	Note for extending the deadline (if at all), Task statistics. Sender link.
TASK CHANGED	RECEIVER	???	???
TASK CHANGED	SENDER	Notification period / deadline / title /. Unclear if necessary.	???

Preliminary User Interface		
GUIDELINES	MINIMALISM	Adopt an approach as minimalistic/frugal/laconic as possible in everything - page elements, actions, text (including in email). As few explanations as possible. Rely on design which will allow the user to just interact, not think. Require as few actions as possible from the users. Additional "super user" actions are made available through "menu" elements. Create as few pages as possible and rely on elements which change to provide additional interface options. All major actions should be achievable through the homepage.
	INSPIRATION	Main source for inspiration is WeTransfer. The lessons which they teach are applied throughout the UI. Ideally, ours will be as simple, smooth, and smart as theirs.
	EMAIL	Email will become the main "gateway" to the website. In the beginning, users will use links from email to navigate and control their tasks.
	IMAGES	Rely on smart icons and pleasing to create an obvious, pleasant, and creatively stimulating experience.

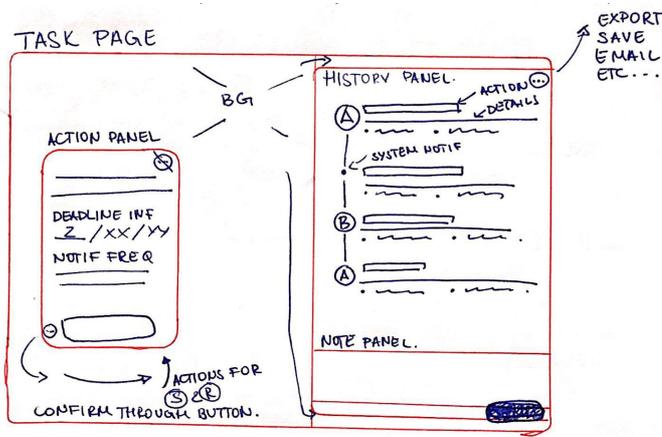


Homepage	
MAIN GOAL	Allow the user to create a task entirely while staying on this page.
BACKGROUND	Fullscreen background. Rotates between artists - simple fade.
MENU/PAGES	3 items: Help, About, Upcoming. All open as panels on homepage. WeTransfer
LINK	There is a link to the artist's page - artistation or similar.
SCROLL	The page is responsive one-screen - there is and should never be a scroll.
MOBILE	Show only the action panel. Burger menu on left. Main goal is the same.



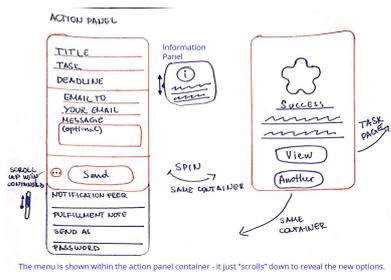
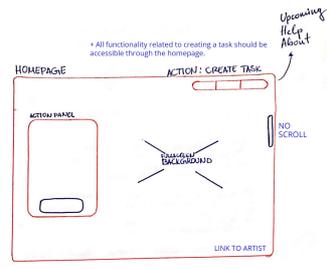
The menu is shown within the action panel container - it just "scrolls" down to reveal the new options.

Action panel	
MAIN GOAL	Collect relevant information from user and send to server.
DYNAMIC	The panel rotates/shifts to represent different steps in the process.
SUCCESS	The panel shifts to a success message after successfully sending information
INFO PANEL	An info panel appears when the user navigates the fields for info/errors.
MENU	The menu contains additional options (can also be configured on task page.)
SEND	The Send button sends the information to the server
VIEW	The View button takes the user to the task page with their Sender URL.
ANOTHER	The Another button flips the panel back to the starting point (cleared)

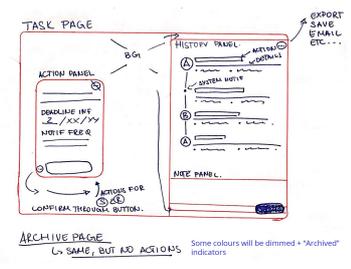


Task page	
MAIN GOAL	Display info about task and provide actions to users.
URL	The page shows different action options based on Sender/Receiver URL ID.
HISTORY	The history panel provides an overview of all actions & notes.
NOTES	The history panel provides an option to leave a plaintext note (supports URL).
BACKGROUND	Same principle as homepage applies. Maybe artist link positioning changes.
ARCHIVE	When the task is archived, the task page becomes an archive page (still same page, same address) - all disabled.

ARCHIVE PAGE
↳ SAME, BUT NO ACTIONS
Some colours will be dimmed + "Archived" indicators



The menu is shown within the action panel container - it just "scrolls" down to reveal the new options.



ARCHIVE PAGE
↳ SAME, BUT NO ACTIONS
Some colours will be dimmed + "Archived" indicators

Homepage	
MAIN GOAL	Allow the user to create a task entirely while staying on this page.
BACKGROUND	Fullscreen background. Rotates between artists - simple fade.
MENU/PAGES	3 items: Help, About, Upcoming. All open as panels on homepage. No Transfer
LINK	There is a link to the artist's page - artistation or similar.
SCROLL	The page is responsive one-screen - there is and should never be a scroll.
MOBILE	Show only the action panel. Burger menu on left. Main goal is the same.

Action panel	
MAIN GOAL	Collect relevant information from user and send to server.
DYNAMIC	The panel rotates/shifts to represent different steps in the process.
SUCCESS	The panel shifts to a success message after successfully sending information
INFO PANEL	An info panel appears when the user navigates the fields for info/errors.
MENU	The menu contains additional options (can also be configured on task page.)
SEND	The Send button sends the information to the server
VIEW	The View button takes the user to the task page with their Sender URL.
ANOTHER	The Another button flips the panel back to the starting point (cleared)

Task page	
MAIN GOAL	Display info about task and provide actions to users.
URL	The page shows different action options based on Sender/Receiver URL ID.
HISTORY	The history panel provides an overview of all actions & notes.
NOTES	The history panel provides an option to leave a plaintext note (supports URL).
BACKGROUND	Same principle as homepage applies. Maybe artist link positioning changes.
ARCHIVE	When the task is archived, the task page becomes an archive page (still same page, same address) - all disabled.

User story & (Corresponding) Front/Backend actions - Follow Along Left to Right, Row by Row			
USER	FRONTEND	BACKEND	
	CREATE A TASK		
Navigate to Homepage	Homepage with focus on Action Panel		

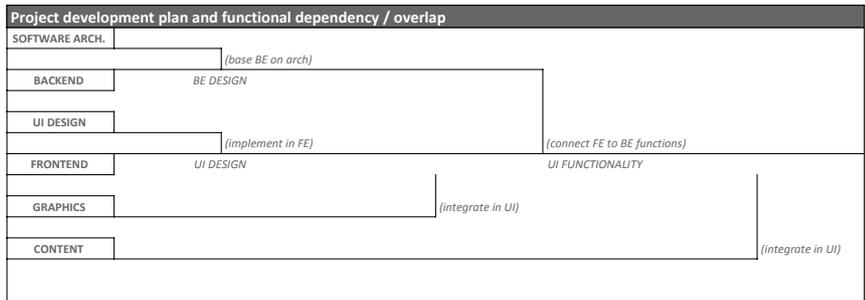
SENDER	Enter info in Action Panel	HOMEPAGE	Homepage with focus on Action Panel.		
	Click Send		Homepage. Show success page of Action Panel.	DB	New task row in DB + Accompanying URLs. Send emails.
	View?		TASK PAGE	Redirect to Task Page for that task ID (With Sender URL)	DB
RECEIVER	Again?	HOMEPAGE	Stay on Homepage. Show primary Action Panel page.		
	Accept task	EMAIL	Receiver clicks a link from an email. Redirected to task page with Receiver URL + code for Accept/Deny.	DB	Edit DB to reflect Receiver choice. Send emails. Update task history.
	Reject task				
VIEW TASK					
SENDER / RECEIVER	Click on link in email (bookmarked / stored in profile ?)	TASK PAGE	Email -> Sender/Receiver link -> Task Page parses URL -> Shows relevant options & functions with correct ID params.		(Only wait to execute functions when called by FE action element)
CLOSE / ABANDON / COMPLETE / EDIT TASK					
SENDER / RECEIVER	Click on the relevant button shown through the Sender / Receiver pages.	TASK PAGE	Parse URL ID -> Show relevant actions -> Action selected -> Call function -> Receive confirmation -> Show confirmation	DB	Edit DB to reflect recent changes, write in history, archive task if applicable, and send out confirmation emails.
WRITE NOTE					
SENDER / RECEIVER	Navigate to History Panel / Note Panel and write plaintext. Click the send button	TASK PAGE	Parse URL ID -> Focus on History Panel -> Text has been entered -> Send button has been pressed -> Send information to DB -> Wait for confirmation -> Query for new data -> Show new data.	DB	Receive new note data + ID -> Write in task history -> send email -> confirm to FE job done. Provide new data for FE to show.
BACKGROUND SWITCHER					
		HOMEPAGE / TASK PAGE	Automatically switch backgrounds after a period of time. Implement efficient loading (bad connections get flat colour background or gradient). Switch only if preload is done.	DB	Provide access to Artist data including image file, link, title, etc.

Backend Components	
SCHEDULER	Creation of, Access to, and Management of all scheduled notifications.
	Time Send out notifications when they are due
DATABASE	Task Metadata
	Name
	Subtask
	Created on
	Deadline
	Notification frequency
	Sender
	Email
	URL
	Receiver
Email	
URL	
FUNCTIONS	Task History
	History Item
	Item title (based on action)
	Content (text)
	Content (action description)
Created on	
Created by	
Create new task	
Edit task data	
Read task data	
Send email	
Add item to history	
Read history data	

Roles and Responsibilities	
SOFTWARE ARCH.	Preliminary done by me
UI DESIGN	Preliminary done by me
BACKEND CODING	Scheduler
	Database
	FE Functions Send Emails (API)
Basic Website Structure	
	General page structure
	Homepage
	Task Page
	Archive mode
	About
	Upcoming
	Help
	General UI panels and components
	Additional pages slide in / appear
	Buttons
Notes input field	
History panel	
History item design	
Action Panel	
	Information Input
	Title
	Task
	Deadline
	Sender
	Email To
	Message
Additional Settings	
Additional Settings Panel Appearance	

FRONTEND CODING		Notification Period
		Fulfillment note (genesis note)
		Send As (Email/Link)
		Password
		Information Panel
		Panel
		Functionality (Dynamic appearance / Anchor)
		Error Checking
		Notifications
		Flipping Functionality
		Flip panel to show new panel
		Success Panel
		UI Design
		Button functionality
		Buttons
	Link up to BE functionality	
	Confirmation functionality	
	Functionality	
	Link up FE with BE functionality	
	Send information from action panel	
	History panel information retrieval	
	Retrieve URLs	
	Post notes	
	Close/Abandon/Edit tasks	
	Show selective functionality based on URL ID	
	Task page	
	Background	
	Connect to artist DB & retrieve data	
	Load efficiently	
	Switch functionality	
	Display artist link	
GRAPHICS	UI Graphics	
		Icons / Explanatory graphics
		Design
		Special panels
	Backgrounds	
	Artists	
	Emails	
CONTENT	Website text	
		UI elements text
		Pages text
		Notifications text
	Emails	

	Work can be commenced without any (or with a few) prerequisites.
	Relevant Green work is required to complete before yellow can begin



To be determined together with you	
SKILLS	Which of the areas described above relate the best to your skillset? Which elements do you feel more comfortable giving to another professional?
THOUGHTS	What do you think about the project and the concept behind it?
CURR. STATE	What is your opinion on the application presented as it is, from a technical standpoint? Does the described above approach have shortfalls?
ARCHITECTURE	What software architecture do you propose for this project? What do you believe is the best way to approach it?
FRAMEWORKS	What frameworks do you propose for this project when you take into account the prototype requirement and the potential scaling up later on?
UI DESIGN	What is your opinion on the UI/UX? What issues do you see in it? Is there room for improvements and originality (if so, where)?
TIME	Estimates about the approximate time it might take you to complete the different stages described above. Arrange them how you see best.
	Developer related questions. Graphics/UI artists and content creators can happily skip them. :)

THANK YOU FOR READING! :)