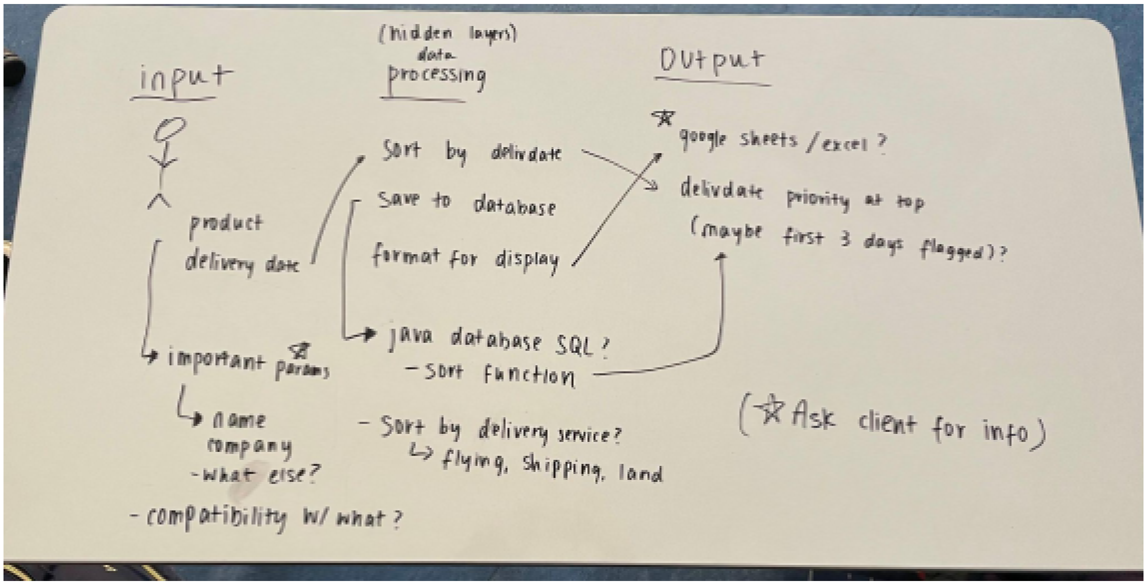
Criterion B: Design

Overall Design

Design 1: Overview



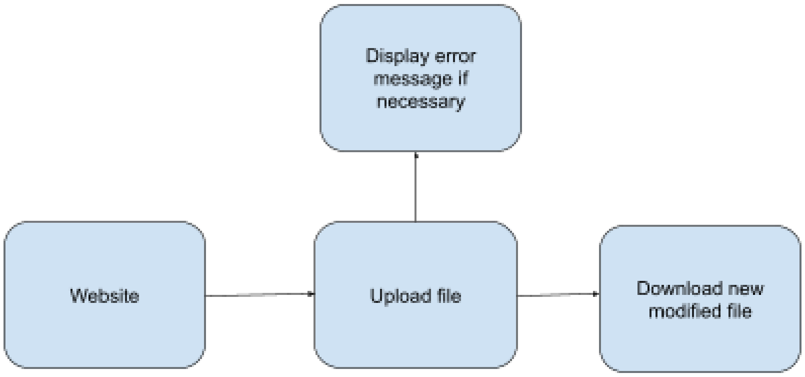
Design 1: App features

Title	Purpose
Log-in page	Ensure that only authorised personnel are able to use the app
Profile page	Users can edit their profile information and change their password, username, email, etc.
Home page	Allow for the user to upload files to the app
Download page	Allow for the user to download the new modified excel file that the database modified to
Error-handling page	To display any error messages that may come out of the file such as not being able to display a certain item or having the item incorrectly placed in the column

Design 2: Overview

Input	Processing	Output
Spreadsheet with variables	Sort based off of the days taken to send and deliver the product	Modified excel file with sorted products in order
(Product --> Dates, ID, Country, etc.)	All stored in an excel file	Also add comments with what kind of shipping is required

Design 2: Website function

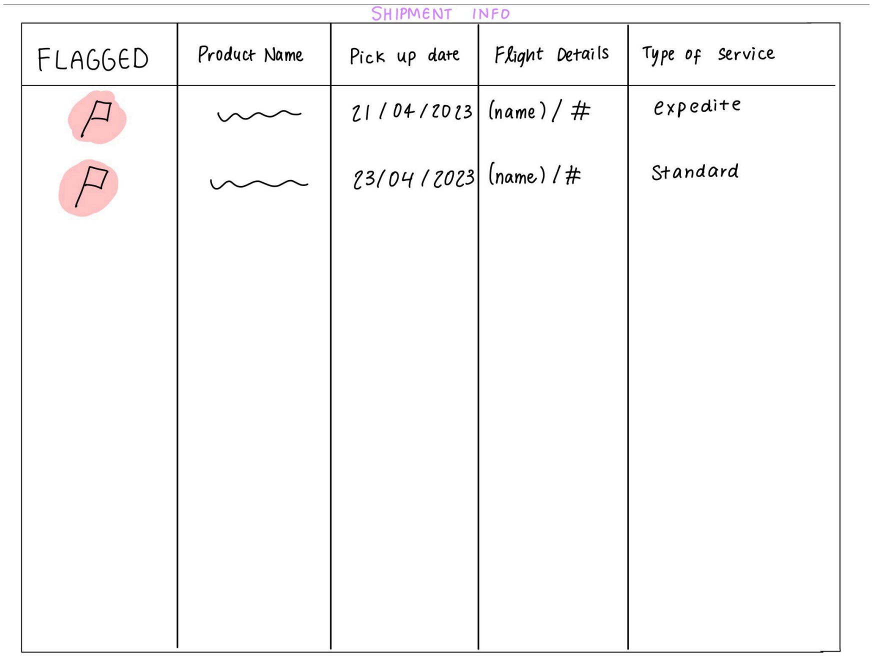

Display error message if necessary
Website
Upload file
Download new modified file

Design 2: Website features

Title	Purpose
Home page	Allow for the user to upload an excel file with the unsorted deliveries
Download page	The user can download the newly modified excel file and be able to open it on microsoft excel
Error-handling page	Display and identify the possible errors in the file that is causing it to be unmodifiable such as wrong variable types or the improper number of columns to be sorted

Mockups

Mockup 1: Spreadsheet formatting (output)



SHIPMENT INFO

FLAGGED	Product Name	Pick up date	Flight Details	Type of service
⚑	〰	21 / 04 / 2023	(name) / #	expedite
⚑	〰	23/04 / 2023	(name) / #	Standard

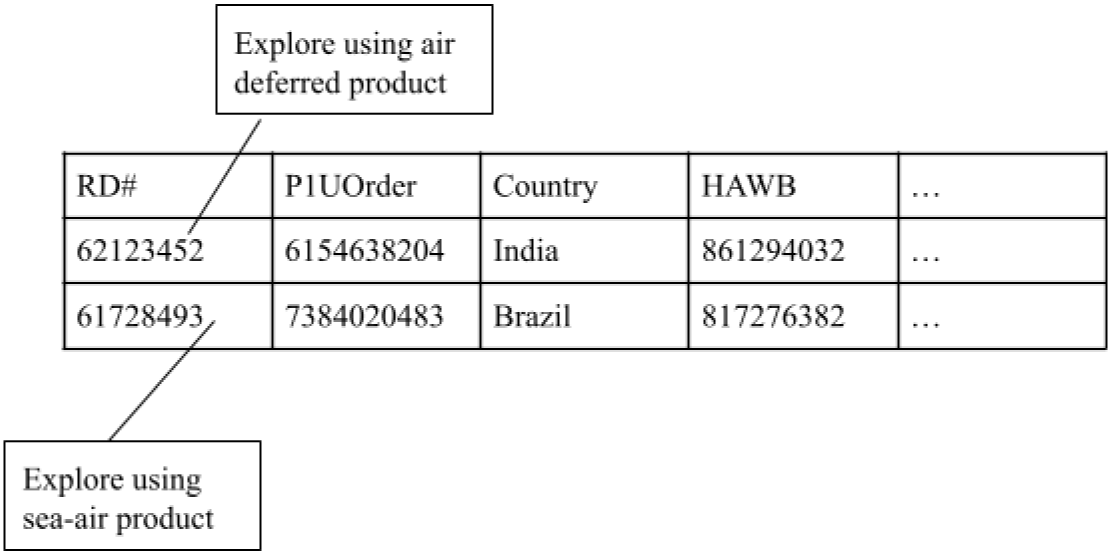
Version 1's unmodified excel file has a flagged column which would allow the user to click on and off on the flags in order to flag and unflag a product.

Mockup 2: Spreadsheet formatting (output)

Flag?	RD#	P1UOrder	Country	HAWB	…
Yes	62123452	6154638204	India	861294032	…
No	61728493	7384020483	Brazil	817276382	…
…	…	…	…	…	…

Version 2 has boolean flag values on the excel file along with different types of column names and types

Mockup 3: Spreadsheet formatting (output)



Explore using air deferred product

RD#	P1UOrder	Country	HAWB	…
62123452	6154638204	India	861294032	…
61728493	7384020483	Brazil	817276382	…

Explore using sea-air product

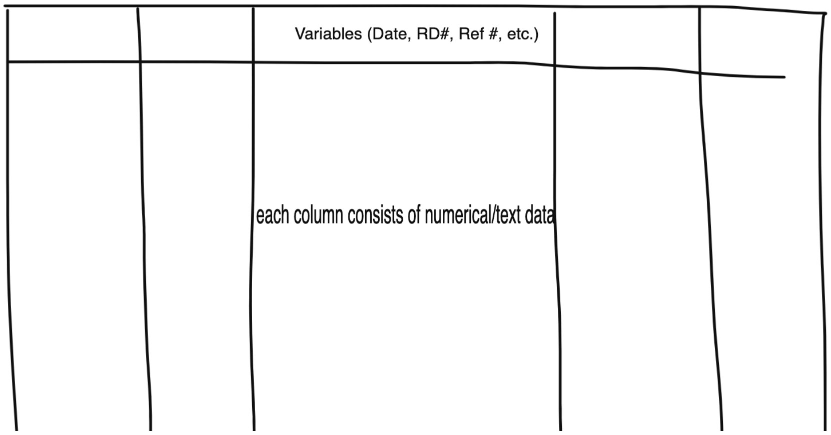
Version has comments displayed on each of the products relating to the type of shipment that should be used instead of flagging them

Mockup 1: Spreadsheet formatting (input)

Product name	Pick up date	Flight details	…
Candle	24/02/23	Lufthansa/LH209	…
Utensils	21/02/23	Japan Airline/JAL123	…
…	…	…	…

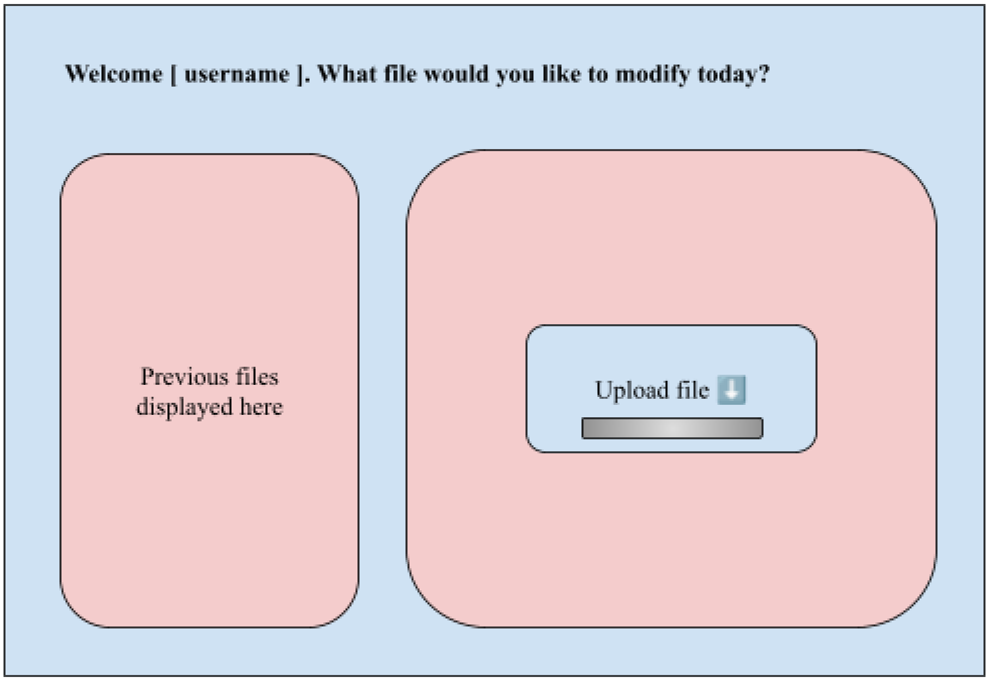
Columns listing details for each of the products. Some of the columns were listed and others needed to be added thus prompting mockup 2.

Mockup 2: Spreadsheet formatting (input)


Variables (Date, RD#, Ref #, etc.)
each column consists of numerical/text data

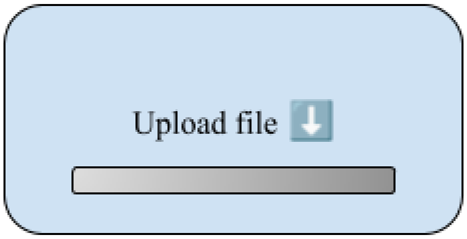
Version 2 has an unordered, input excel spreadsheet with variable names in the first row of their respective columns while data sits in its prescribed column.

Mockup 1: Home page


Welcome [username]. What file would you like to modify today?

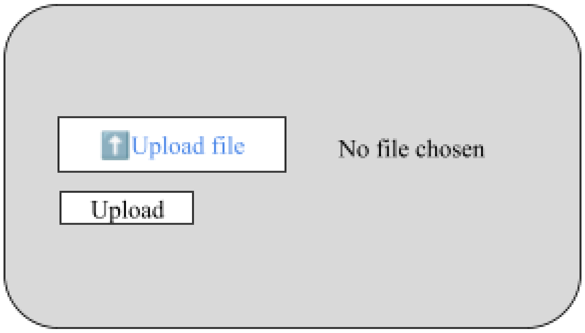
Previous files that have been uploaded/modified would be on the left-hand side for further reference while the new file upload would be on the right hand side.

Mockup 2: Home page



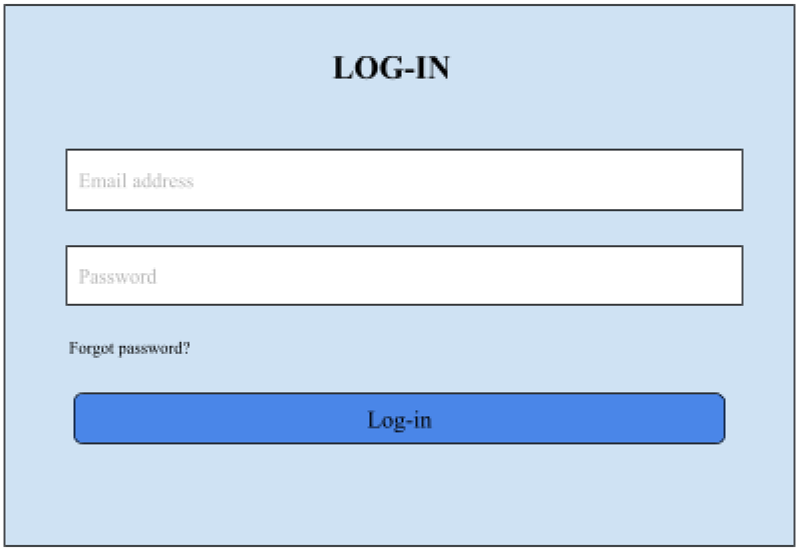
More simplified website design

Mockup 3: Home page



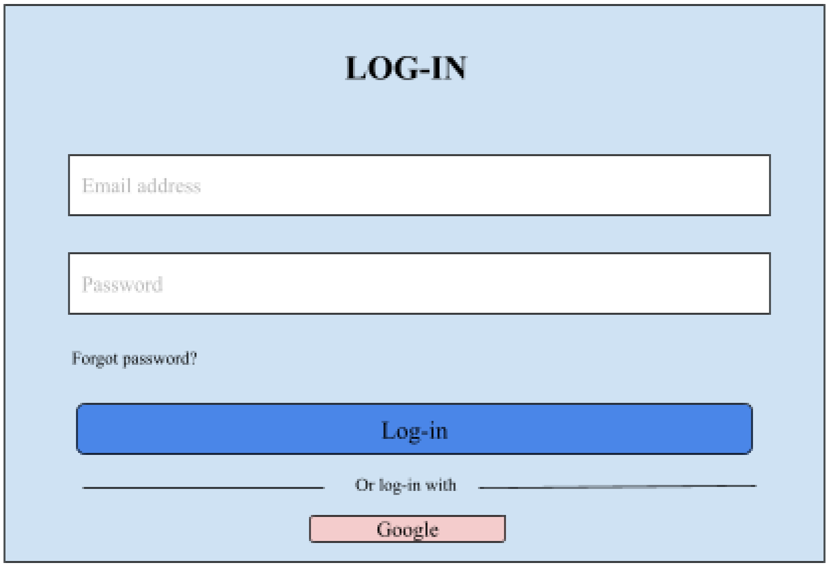
Final version of the home page

Mockup 1: Log-in page



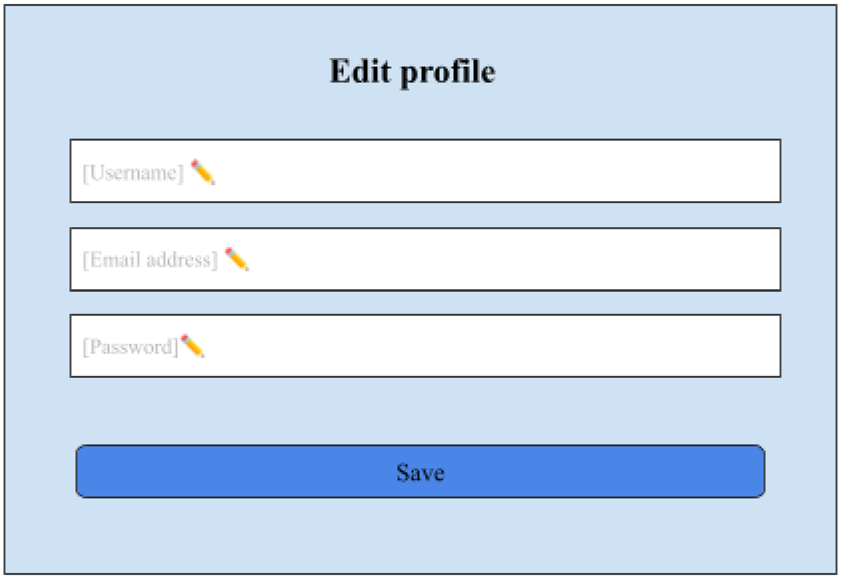
Version 1 of log-in page with company provided email address and password

Mockup 2: Log-in page

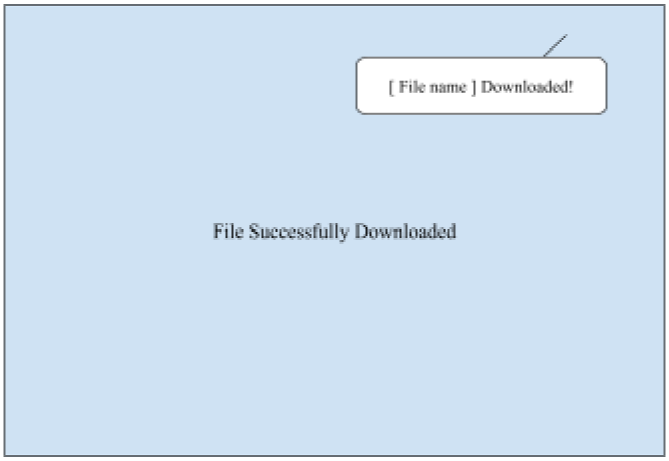


Version 2 gives the user the option to sign in with google. This is more preferred as google has 2 factor authentication.

Mockup: 1: Profile page

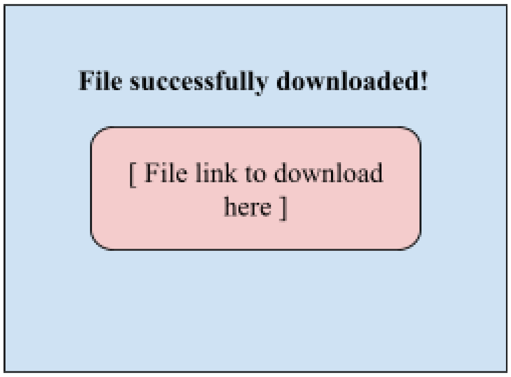


Mockup 1: Download Page



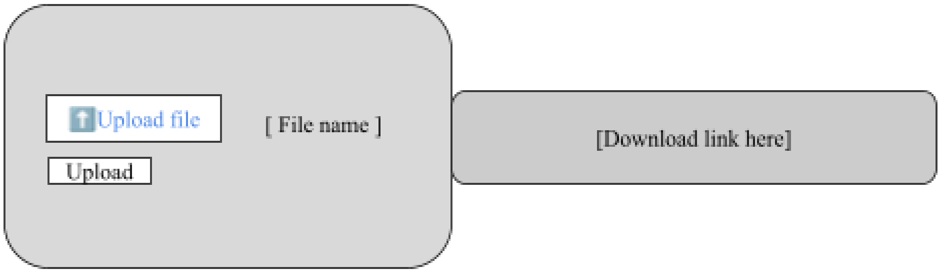
Version 1 immediately downloads the file into the computer's desktop

Mockup 2: Download page



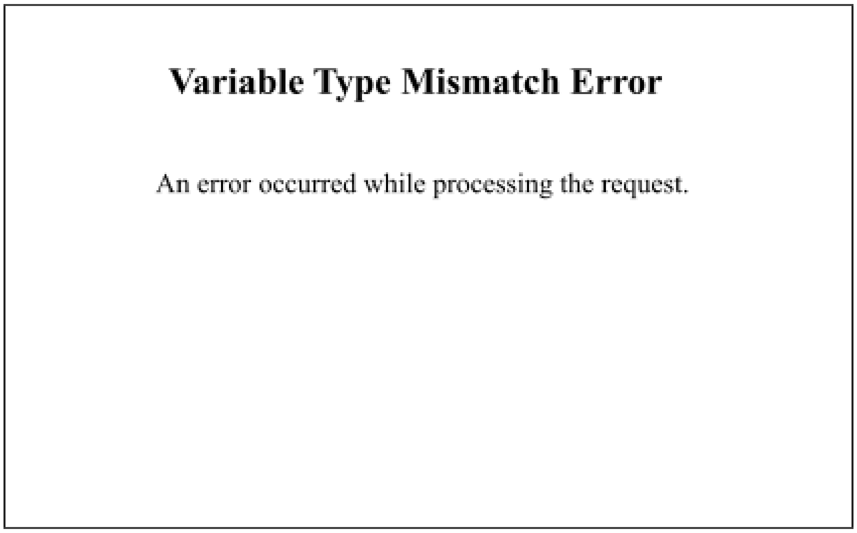
Version 2 requires a click-to-download feature

Mockup 3: Download page



Mockup 3 contains the same click-to-download feature as mockup 2, however it is all done on the same page as the initial upload.

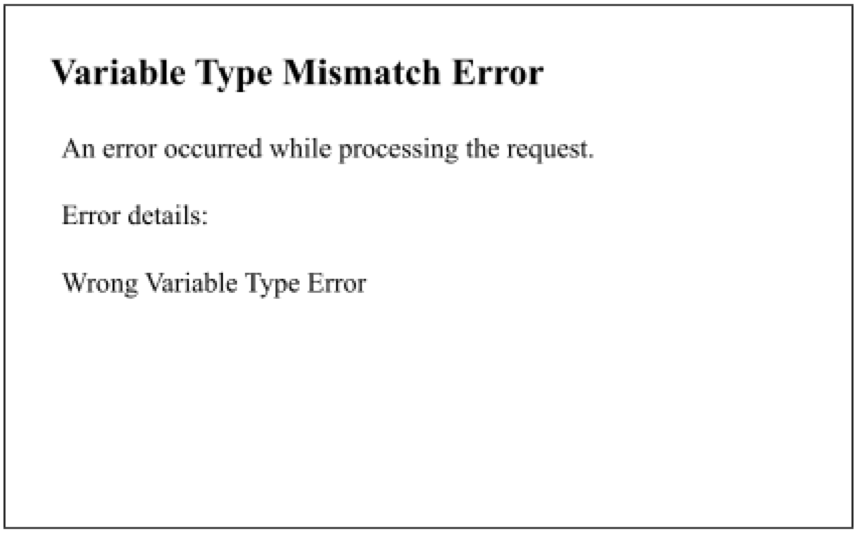
Mockup 1: Error-handling page



Variable Type Mismatch Error

An error occurred while processing the request.

Mockup 2: Error-handling page



Variable Type Mismatch Error

An error occurred while processing the request.

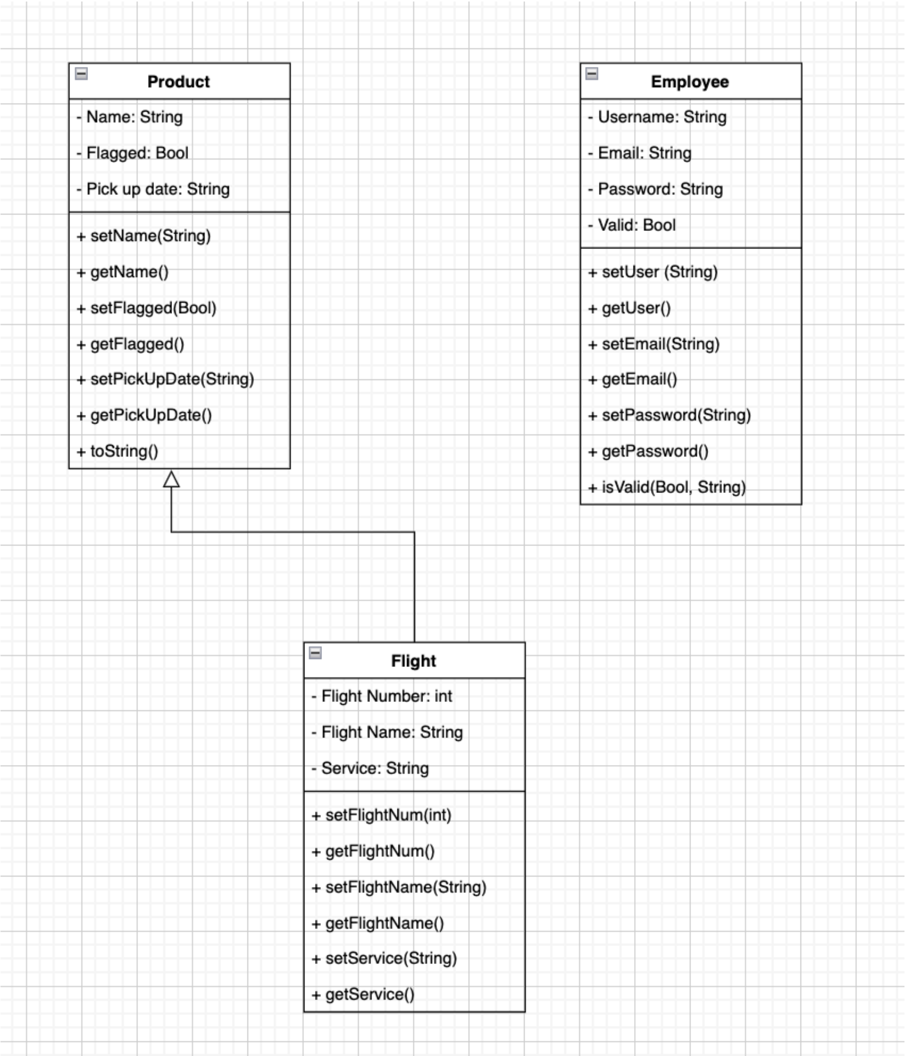
Error details:

Wrong Variable Type Error

Mockup 2 consists of more details that describe what error is occurring

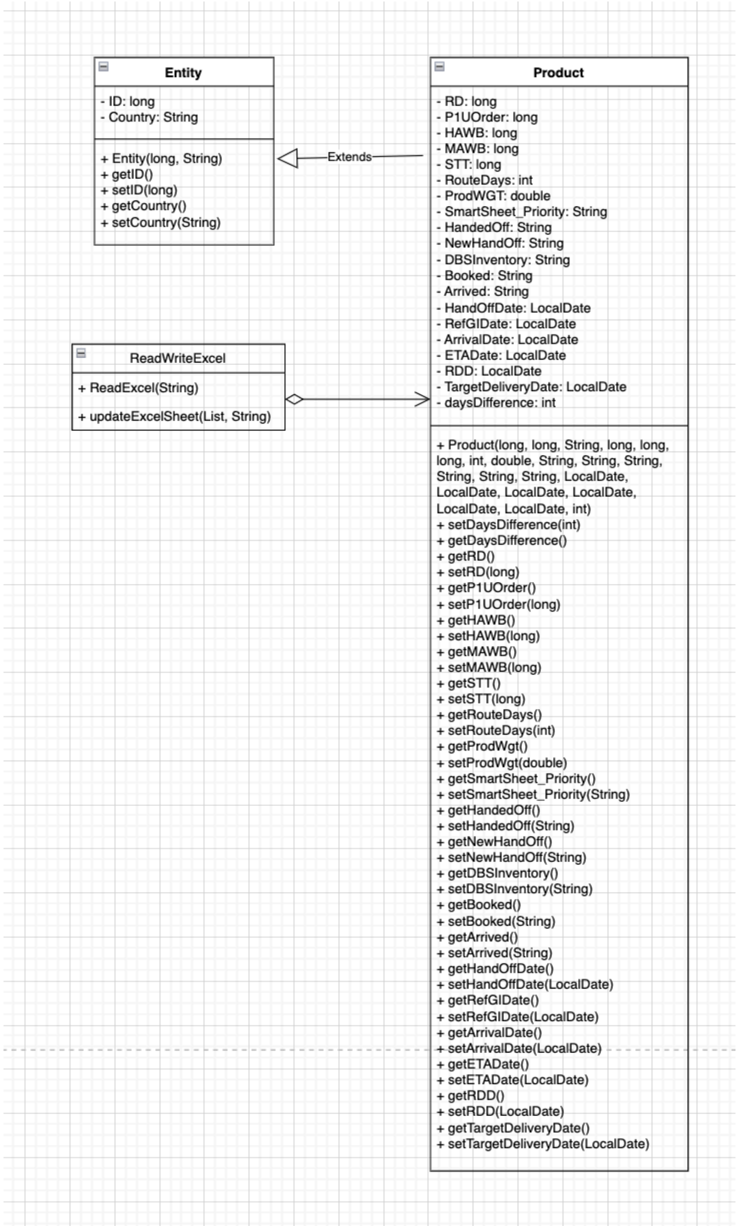
UML Diagrams

Design 1: UML Diagram



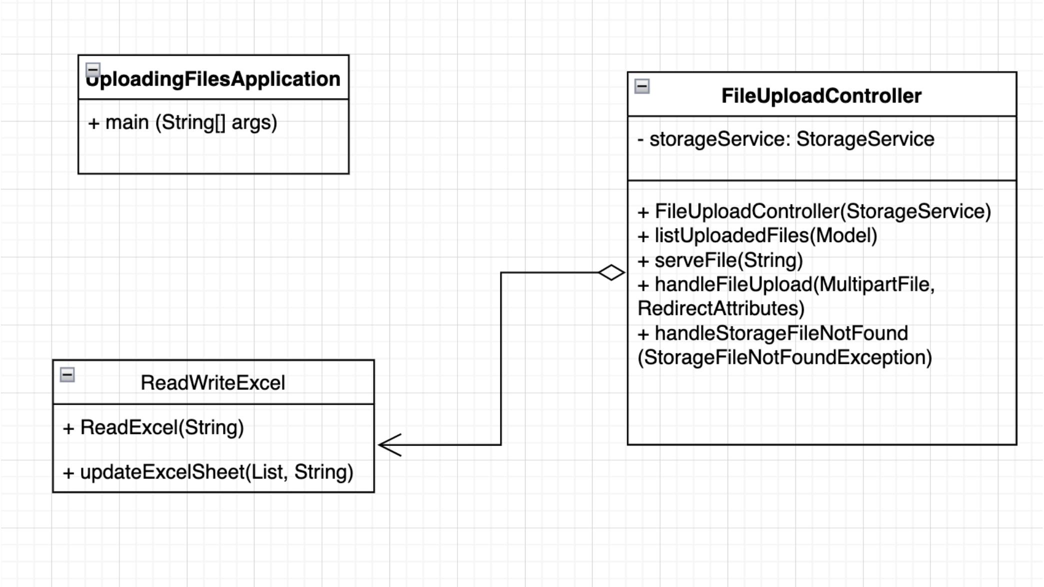
Classes being used in the first design. All of these classes were rewritten afterwards to suit the new design (Design 2)

Design 2: UML Diagram (package - properties)



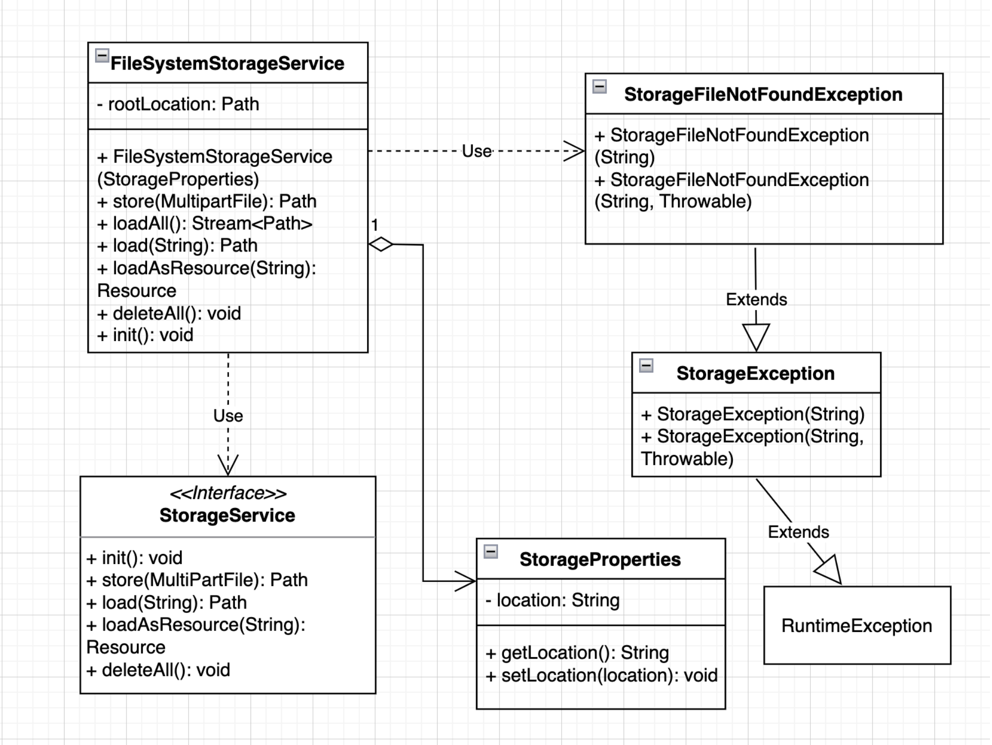
UML Diagram for the classes in package “properties”

Design 2: UML Diagram (package - uploadingfiles)


UploadingFilesApplication
+ main (String[] args)
FileUploadController
- storageService: StorageService
+ FileUploadController(StorageService)
+ listUploadedFiles(Model)
+ serveFile(String)
+ handleFileUpload(MultipartFile, RedirectAttributes)
+ handleStorageFileNotFound (StorageFileNotFoundException)
ReadWriteExcel
+ ReadExcel(String)
+ updateExcelSheet(List, String)

UML Diagram for the classes in package “uploading files”

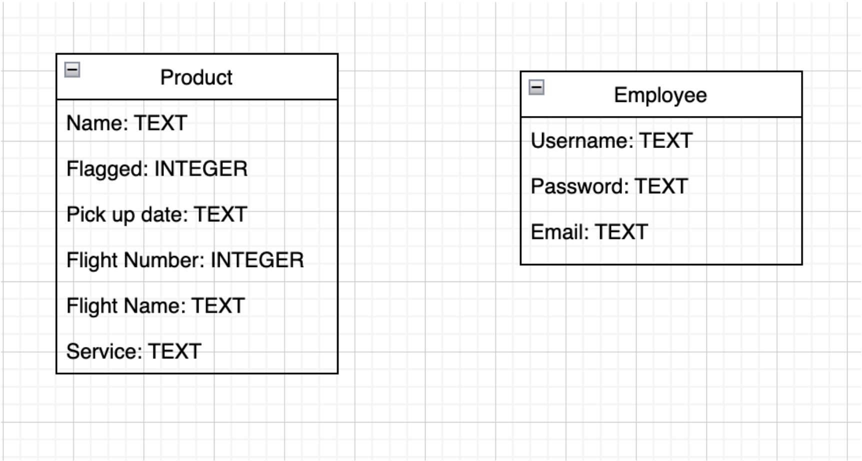
Design 2: UML Diagram (package - storage)


FileSystemStorageService
- rootLocation: Path
+ FileSystemStorageService (StorageProperties)
+ store(MultipartFile): Path
+ loadAll(): Stream<Path>
+ load(String): Path
+ loadAsResource(String): Resource
+ deleteAll(): void
+ init(): void
Use
StorageFileNotFoundException
+ StorageFileNotFoundException (String)
+ StorageFileNotFoundException (String, Throwable)
1
Extends
StorageException
+ StorageException(String)
+ StorageException(String, Throwable)
Use
<<Interface>>
StorageService
+ init(): void
+ store(MultiPartFile): Path
+ load(String): Path
+ loadAsResource(String): Resource
+ deleteAll(): void
Extends
StorageProperties
- location: String
+ getLocation(): String
+ setLocation(location): void
RuntimeException

UML Diagram for the classes in package “storage”

Database Structure

Design 1: SQLite Database Structure



Class Dictionaries

Design 1: Class Dictionary

Product class

Methods

Method Name	Signature	Description
Product()	Product(n: String, f: Bool, p: String)	Constructor with arguments
getName()	getName(): String	Returns name of product
setName()	setName(n: String)	Sets new name of product
getFlagged()	getFlagged(): Bool	Returns if flagged or not
setFlagged()	setFlagged(f: Bool)	Sets flagged for product
getPickUpDate()	getPickUpDate(): String	Returns pick-up-date of product
setPickUpDate()	setPickUpDate(p: String)	Sets pick-up-date
toString()	toString(): String	Returns all the properties of the product object

Properties

Property Name	Type	Description
Name	String	The name of the product
Flagged	Bool	Tells the user if it’s a priority object or not based on the date of pick up
Pick up date	String	The date when the product is being picked up from delivery

Flight class

Methods

Method Name	Signature	Description
Flight()	Flight(num: int, name: String, s: String)	Constructor with arguments
getFlightNumber()	getFlightNumber(): int	Returns flight number
setFlightNumber()	setFlightNumber(num: int)	Sets flight number
getFlightName()	getFlightName(): String	Returns flight name
setFlightName()	setFlightName(name: String)	Sets flight name
getService()	getService(): String	Returns name of service
setService()	setService(s: String)	Sets name of service

Properties

Property Name	Type	Description
Flight Number	Integer	Flight number detail
Flight Name	String	Name of airline
Service	String	Type of service that the package is being shipped through (eg. express)

Employee class

Methods

Method Name	Signature	Description
Employee()	Employee(u: String, e: String, p: String, v: Bool)	Constructor with arguments
getUsername()	getUsername(): String	Returns username
setUsername()	setUsername(u: String)	Sets username
getEmailAddress()	getEmailAddress(): String	Returns email address
setEmailAddress()	setEmailAddress(String)	Sets email address
getPassword()	getPassword(): String	Returns password
setPassword()	setPassword(String)	Sets password
getValid()	getValid(): Bool	Returns if valid or not
setValid()	setValid(Bool)	Sets valid or not

Properties

Property name	Type	Description
Username	String	Username of the employee signing in
Email address	String	Email address to reset password.
Password	String	Password for the employee signing in
Valid	Boolean	Sees if password meets certain requirements (certain length, has certain characters)

Design 2: Class Dictionary

Product class

Methods

Method Name	Signature	Description
Product()	Product(RD: long, P1UOrder: long, country: String, HAWB: long, MAWB: long, STT: long, RouteDays: int, ProdWgt: double, SmartSheet_Priority: String, HandedOff: String, NewHandOff: String, DBSInventory: String, Booked: String, Arrived: String, HandOffDate: LocalDate, RefGIDate: LocalDate, ArrivalDate:LocalDate, ETADate: LocalDate, RDD: LocalDate, TargetDeliveryDate: LocalDate, daysDifference: int)	Constructor with arguments
setdaysDifference()	setdaysDifference(daysDifference: int)	Sets daysDifference
getDaysDifference()	getDaysDifference(): int	Returns daysDifference
setRD()	setRD(RD: long)	Sets RD number
getRD()	getRD(): long	Returns RD number
setP1UOrder()	setP1UOrder(P1UOrder: long)	Sets P1UOrder number
getP1UOrder()	getP1UOrder(): long	Returns P1UOrder number
setHAWB()	setHAWB(HAWB: long)	Sets HAWB number
getHAWB()	getHAWB(): long	Returns HAWB number
setMAWB()	setMAWB(MAWB: long)	Sets MAWB number

getMAWB()	getMAWB(): long	Returns MAWB number
setSTT()	setSTT(STT: long)	Sets STT number
getSTT()	getSTT(): long	Returns STT number
setRouteDays()	setRouteDays(RouteDays: int)	Sets number of days in the route
getRouteDays()	getRouteDays(): int	Returns the number of days in the route
setProdWgt()	setProdWgt(ProdWgt: double)	Sets the product weight
getProdWgt()	getProdWgt(): double	Returns the weight of the product
setSmartSheet_Priority()	setSmartSheet_Priority(SmartSheet_Priority: String)	Sets priority of smartsheet
getSmartSheet_Priority()	getSmartSheetPriority(): String	Returns priority of smartsheet
setHandedOff()	setHandedOff(HandedOff: String)	Sets whether it was handed off or not
getHandedOff()	getHandedOff(): String	Returns whether handed off or not
setNewHandOff()	setNewHandOff(NewHandOff: String)	Sets the new hand off status
getNewHandOff()	getNewHandOff(): String	Returns the new hand off status
setDBSInventory()	setDBSInventory(DBSInventory: String)	Sets whether the status of the product in DBS inventory or not
getDBSInventory()	getDBSInventory(): String	Returns whether the product is in DBS inventory or not
setBooked()	setBooked(Booked: String)	Sets whether the product is booked or not
getBooked()	getBooked(): String	Returns whether the product is booked or not

setArrived()	setArrived(Arrived: String)	Sets arrival status of product
getArrived()	getArrived(): String	Returns whether the product has arrived or not
setHandOffDate()	setHandOffDate(HandOffDate: LocalDate)	Sets the date of hand off for product
getHandOffDate()	getHandOffDate(): LocalDate	Returns the date of hand off for product
setRefGIDate()	setRefGIDate(RefGIDate: LocalDate)	Sets the RefGIDate for product
getRefGIDate()	getRefGIDate(): LocalDate	Returns the RefGIDate for the product
setArrivalDate()	setArrivalDate(ArrivalDate: LocalDate)	Sets arrival date for the product
getArrivalDate()	getArrivalDate(): LocalDate	Returns the arrival date for the product
setETADate()	setETADate(ETADate: LocalDate)	Sets the ETADate for the product
getETADate()	getETADate(): LocalDate	Returns the ETADate of the product
setRDD()	setRDD(RDD: LocalDate)	Sets the RDD for the product
getRDD()	getRDD(): LocalDate	Returns the RDD for the product
setTargetDeliveryDate()	setTargetDeliveryDate(TargetDeliveryDate: LocalDate)	Sets the target delivery date of the product
getTargetDeliveryDate()	getTargetDeliveryDate(): LocalDate	Returns the target delivery date of the product

Properties

Property Name	Type	Description
RD	Long	Reference number
P1UOrder	Long	Order number

HAWB	Long	HAWB number
MAWB	Long	MAWB number
STT	Long	STT number
RouteDays	Int	Number of days on route
ProdWgt	Double	Weight of product
SmartSheet_Priority	String	Priority on SmartSheet
HandedOff	String	Hand off status of product
NewHandOff	String	New hand off status
DBSInventory	String	DBS Inventory status
Booked	String	Booked status of product
Arrived	String	Arrival status of product
HandOffDate	LocalDate	Date of hand off
RefGIDate	LocalDate	Date of RefGI
ArrivalDate	LocalDate	Date of arrival
ETADate	LocalDate	Date of ETA
RDD	LocalDate	Date of RDD
TargetDeliveryDate	LocalDate	Date of TargetDelivery

Entity Class

Methods

Method Name	Signature	Description
Entity()	Entity(ID: long, Country: String)	Constructor with arguments
getID()	getID(): long	Returns ID of entity
setID()	setID(ID: long)	Sets ID of entity
getCountry()	getCountry(): String	Returns country of entity
setCountry()	setCountry(Country: String)	Sets country of entity

Properties

Property name	Type	Description
ID	long	ID of entity
Country	String	Country of entity

Readwriteexcel class

Methods

Method Name	Signature	Description
ReadExcel()	ReadExcel(filePath: String): List	Reads excel file and stores all of the cells in their corresponding variable names
updateExcelSheet()	updateExcelSheet(products: List, filePath: String)	Sorts the rows based on their dates and then stores it back into the modified excel file

FileUploadController class

Methods

Method Name	Signature	Description
FileUploadController()	FileUploadController(storage Service: StorageService)	Sets up storage service
ListUploadedFiles()	ListUploadedFiles(model: Model): String	List uploaded files into the server
serveFile()	serveFile(filename: String): ResponseEntity	Loading from storageService based on filename and returns as part of HTTP response with appropriate headers for file download
handleFileUpload()	handFileUpload(file: MultipartFile, redirectAttributes: RedirectAttributes): String	Makes sure that the file is uploaded successfully and throws an error if not successful
handleStorageFileNotFound()	handleStorageFileNotFound (exc:StorageFileNotFoundException): ResponseEntity	Handles situations where the requested file is not found

Properties

Property name	Type	Description
storageService	StorageService	Object that manages storage

UploadingFilesApplication class

Methods

Method name	Signature	Description
init()	init(storageService: StorageService)	Initialises and deletes files in the main application
main()	main(String[] args)	Responsible for running the whole application

FileSystemStorageService class

Methods

Method name	Signature	Description
FileSystemStorageService()	FileSystemStorageService(properties: StorageProperties)	Sets up the root location for the file
store()	store(file: MultiPartFile): Path	Stores the file
loadAll()	loadAll(): Stream	Reads the stored file and loads them
load()	load(filename: String): Path	Resolves the uploaded file
loadsAsResource()	loadAsResource(filename: String): Resource	Locates the file and checks if it’s readable or not
deleteAll()	deleteAll()	Deletes the file
init()	init()	Initialises storage

Properties

Property name	Type	Description
rootLocation	Path	Stores the root location of the file

StorageException class

Methods

Method name	Signature	Description
StorageException()	StorageException(message: String)	Constructor for storage exception error message
StorageException()	StorageException(message: String, cause: Throwable)	Constructor for storage exception error message and cause

StorageFileNotFoundException class

Methods

Method name	Signature	Description
StorageFileNotFoundException()	StorageFileNotFoundException(message: String)	Constructor for storage file not found exception message
StorageFileNotFoundException()	StorageFileNotFoundException(message: String, cause: Throwable)	Constructor for storage file not found exception message and cause

StorageProperties class

Methods

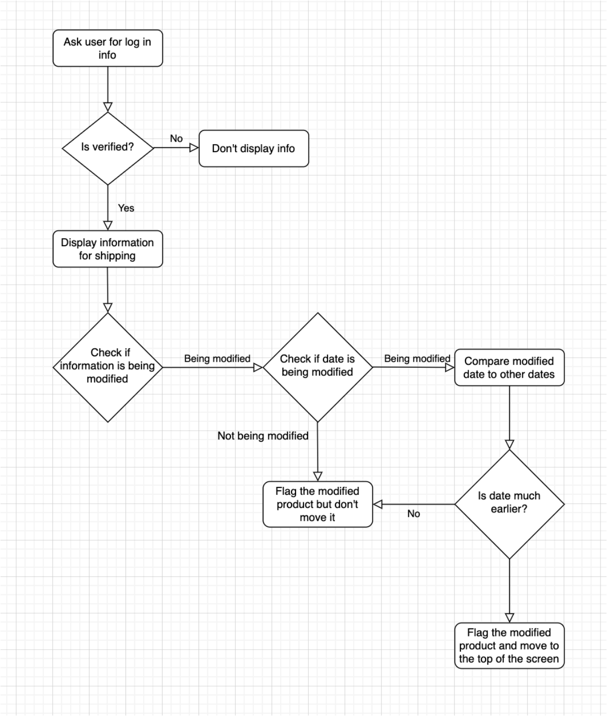
Method name	Signature	Description
getLocation()	getLocation(): String	Returns location of storage
setLocation()	setLocation(location: String)	Sets the location of storage

Properties

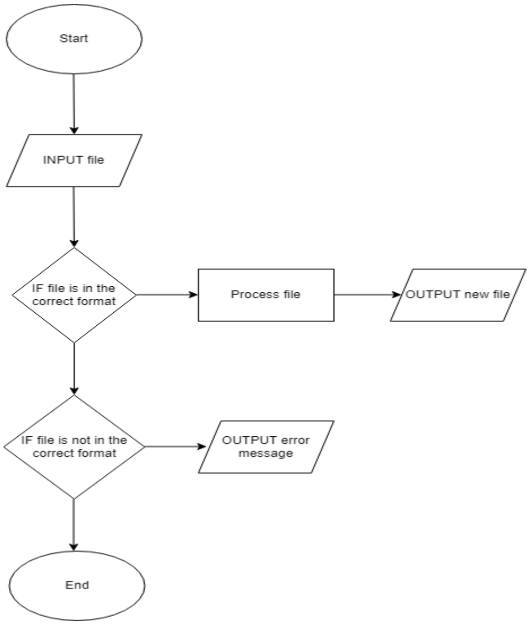
Property name	Type	Description
location	String	Stores the location of storage

Algorithms

Design 1: Overview flowchart


Ask user for log in info
Is verified?
No
Don't display info
Yes
Display information for shipping
Check if information is being modified
Being modified
Check if date is being modified
Being modified
Compare modified date to other dates
Not being modified
Flag the modified product but don't move it
No
Is date much earlier?
Flag the modified product and move to the top of the screen

Design 2: Overview flowchart


Start
INPUT file
IF file is in the correct format
Process file
OUTPUT new file
IF file is not in the correct format
OUTPUT error message
End

Log-in page 1

If (email != valid) → Prompt entering of new email
If (password != valid) → prompt entering of new password
If (password & email = valid & Log-in button = pressed) → redirect to home page
If (forgot password = pressed) → redirect to page that allows for password to be sent to email address

Log-in page 2

If (email != valid) → Prompt entering of new email
If (password != valid) → prompt entering of new password
If (password & email = valid & log-in button = pressed) → redirect to home page
If (forgot password = pressed) → redirect to page that allows for password to be sent to email address
If (google button = pressed) → redirect to google sign in page

Profile Page 1

If (username pencil icon = pressed) → Change username
If (email address pencil icon = pressed) → Change email address
If (password pencil icon = pressed) → Change password
If (save = pressed) → Allow for all of the changes to be saved
If (save != pressed) → Don't allow for the username, email address or password to be saved and revert back to old form

Download page 1

If (file upload = true) → Immediately download onto desktop
If (file upload = false) → Redirect to error page → display error message
If (download = pressed) → Open onto excel in desktop

Download page 2

If (file upload = true) → Redirect to new file download page
If (file upload = false) → Redirect to error page → display error message
In new file download page:

```
Display → “File successfully downloaded!”
If (file link download = pressed) → Download file onto desktop
```

Download page 3

```
If (file upload = true) → Display download link
If (file upload = false) → Redirect to error page → display error message
If (file link download = pressed) → Download file onto desktop
```

Error handling page 1

```
Find the type of error → display type of error + “error” → display “an error occurred while processing the request”
```

Error handling page 2

```
Find the type of error → display type of error + “error”
Display → “An error occurred while processing the request”
Display → “Error details: ”
Display → Detail of error
```

Test Plan

Success Criterion	**Method of Testing**	**Expected Results**
Support for iOS and Windows	Alpha testing/Beta testing	Works on both platforms and client will provide feedback on how it performed on both platforms
Be able to upload an excel file with the right rows and columns on it to the website	Alpha testing	*Abnormal:* The excel file with one wrong row is uploaded to the website *Expected result:* Error is displayed and the file won't be returned *Abnormal:* The excel file with one wrong column is uploaded to the website *Expected result:* Error is displayed and the file won't be returned *Normal:* Excel file is correctly formatted and uploaded to the website *Expected result:* File will be modified and returned successfully *Extreme:* The excel file with the wrong row and columns is uploaded to the website *Expected result:* Error is displayed and the file won't be returned *Extreme:* The excel file with multiple wrong rows or columns is uploaded to the website *Expected result:* Error is displayed and file won't be returned

Be able to successfully store the variables from the excel file in the program and able to process it correctly.	Alpha testing	*Abnormal:* One wrong variable is uploaded into the excel file *Expected result:* Display error message and file will not process at all *Normal:* Right variables are uploaded into the excel file *Expected result:* Display the correctly modified file and allow for download *Extreme:* Multiple wrong variables are uploaded into the excel file *Expected result:* Display error message and file will not process at all
Be able to sort the data into shipping-priority categories based on the difference between the HandOffDate and TargetDeliveryDate.	Alpha testing	*Abnormal:* A few unsorted AND sorted HandOffDate and TargetDeliveryDate products are stored in the file *Expected result:* Modified file will contain the correct sorting of products based on these differences in dates *Normal:* Upload file with some unsorted and/or some sorted number of days between HandOffDate and TargetDeliveryDate *Expected result:* Modified file will contain the correct sorting of products based on these differences in dates *Extreme:* All unsorted number of days between HandOffDate and TargetDeliveryDate *Expected result:* Modified file will contain the correct sorting of products based on

		these differences in dates
Rewriting the sorted products back to the excel file	Alpha testing	*Abnormal:* A few sorted and unsorted products will be in the original file *Expected result:* Display error message and the products will not be sorted and thus the file will not have the correct information outputted *Normal:* The sorted products are processed correctly *Expected result:* The modified excel file will have the correct information outputted *Extreme:* Multiple sorted and unsorted products will be in the original file *Expected result:* Display error message and the products will not be sorted and thus the file will not have the correct information outputted
Write a comment about which shipping category the product is in based on the difference in day	Alpha testing/Beta testing	*Abnormal:* Some of the products may have the same difference in day and others will be different *Expected result:* The modified excel file will correctly write a comment for the shipping category based on the difference in day *Normal:* The products are mostly different in terms of difference in day *Expected result:* The modified excel file will correctly write a comment for the shipping category based on the difference in day

		Extreme: The products are all different in terms of difference in day *Expected result:* The modified excel file will correctly write a comment for the shipping category based on the difference in day *Extreme:* The products are all the same in terms of difference in day *Expected result:* The modified excel file will correctly write a comment for the shipping category based on the difference in day
Be able to open the modified excel file and see all of the products sorted	Alpha testing/Beta testing	*Abnormal:* Products are not sorted *Expected result:* The modified excel file is not modified and instead displays an error message relating to any of the test cases above *Normal:* The file is correctly sorted *Expected result:* The modified excel file is correctly modified and easy to view with the correct comments in the right place
Be able to show an error when the variable type is not correct	Alpha testing/Beta testing	*Abnormal:* One variable is incorrect *Expected result:* File is not returned and the error message for that specific variable is not shown *Normal:* Multiple variables are incorrect (most likely in a column) *Expected result:* File is not returned and error message is

		displayed for that specific variable

Record of tasks

Record of Tasks

Works Consulted

"DB Browser for Sqlite." *DB Browser for SQLite*, 4 May 2023, https://sqlitebrowser.org/.