

# **SplitPay: Payments Ledger**

**FINAL REVIEW REPORT**

**Submitted by**

**Tanmay Bansal(19BCE0421)**

**Atul Agarwal (19BCE0436)**

**Saurav Sharma(19BCE2218)**

**Nisheta Gupta 19BCE2233)**

**Pranavi Pendyala(18BCE2253)**

**Prepared For**

**Human Computer Interaction (CSE4015)**

**PROJECT COMPONENT**

**Submitted To**

**Ragunthar . T**

**Assistant Professor (Senior)**

**School of Computer Science and Engineering**



## Table of Contents

- 1. Abstract**
- 2. Introduction**
- 3. Requirement Analysis (Data Collected)**
- 4. Data Flow (HTA, Story board)**
- 5. 10 Heuristic evaluation & Shneiderman's 8 Golden Rules matching with UI**
- 6. Testing**
- 7. Implementation**
- 8. Conclusion**

## **1. Abstract**

With the advent of technology and personal smartphones, many real-world avenues have turned to technical solutions like apps and websites. One such field is that of Banking and Payments. In this day in age a lot of people are using smartphones and nowadays every person wants their smartphone to have all the applications that can help them in everyday life, one of those applications that we have seen has risen in use are the Payments app or E-wallets. We have seen payments apps like PayTM, Google Pay and Phone Pay come to light in recent years. As students of a university, we don't like to use cash as the primary way of paying either shopkeepers or our friends so we are usually using these E-wallets and Payments App as the primary source of spending our money. However, we face difficulties when it comes to splitting bills, which is the case many times.

So, we look to apply a user-centred design approach to develop an e-ledger application which not only supports maintaining payment records but also lets a group of users split a payment amongst themselves, and settle them at their own ease, within the app.

## **2.Introduction**

We have developed an interface to allow users, mainly university students, to record and maintain payments ledgers to avoid and resolve any conflicts regarding the settlement of payments. With this application we aim to provide an easy and comprehensive solution to any payment settlement problems. Users can choose to record a single payment to a person or choose to split it, a unique feature of the app.

One unique feature that we have added to this application which is not present in any of the applications present in current day is A SplitPay - functionality. In this SplitPay functionality users can split their bills among their friends or colleagues, i.e. if more than one people have purchased some amount of goods, for e.g., food, then they have the ability to split their bill according to the amount of food that each person eats.

### **3.REQUIREMENT ANALYSIS**

The first step of the project was to employ two kinds of data collection methods to understand the needs of our primary stakeholders better. We chose to conduct User Interviews and an Online Survey.

#### User Interviews

The interviews helped us understand the unique behaviours of a small number of people which let us find specific details about the way they perceive the current online payment platforms. We conducted user interviews with 6 people. We reached out to our friends/classmates to gain insights about the processes and mechanisms they follow while paying bills and splitting expenses.

The following is the list of questions we asked during these interviews:

1. What are the online payments apps you have experience of using? Tell me about the last time you used one.
2. What was the kind of setting? Were you with friends or alone?
3. Do you face any issues when using an payment app? What are they?
4. Do you prefer using a payments app over paying Cash or by Card?
5. What is the scale of amounts that you regularly pay? Would you use a payment app to transfer larger amounts of money?
6. Do you trust these apps to keep your money safe?
7. Would you recommend someone else to use a payments app in day to day life? Why?
8. When was the last time that you can remember splitting a bill? Tell me about that experience.
9. How many of you were there when you split the bill?
10. Do you frequently split bills with these same people?
11. What is the way in which you split? Do you use any tools? Tell me the process that goes into it.
12. Do you split equally amongst all, by what each person has consumed, or does it vary from time to time?
13. Are you aware about any platforms or mechanisms which can help you split bills with people?

These were semi-structured interviews, so we often cross-questioned the interviewees on the basis of their answers.

In summary, the following are the insights we gained from these interviews:

- People generally found e-wallets and online payment as a more convenient means of payment as compared to more conventional ones. All of them would recommend the use of such platforms for day-to-day expenses. This means there is a good scope for the usage of such a platform.

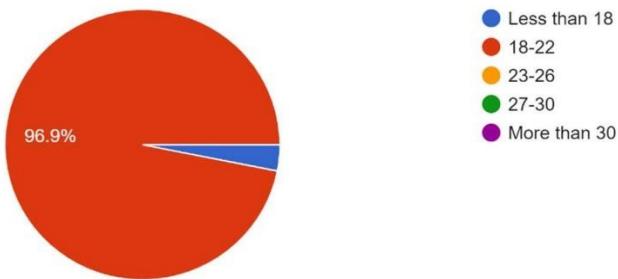
- Their major issues with such apps are concerned with server errors. While most are satisfied with the usability of the big scale apps, various apps come with various difficulties in setting up these apps and linking them with other services. Some smaller scale apps have some usability issues as well. So, it is very important to have a highly usable design to ensure user satisfaction. People
- People generally find themselves using Cash or Card for larger amounts due to more security during the payment process. One of the interviewees also voiced concern as he may not be sure if the large amount has been transferred. Still many are satisfied with the security measures in a payment app when it comes to smaller amounts.
- One of the six interviewees very rarely uses such apps. He voiced his reasons as cash being reliable enough for him. He usually uses such apps in social situations where expenses need to be split.
- All the interviewees told us that they find themselves splitting bills in social situations, like when they meet for meals, when they travel, etc. So, there is always a need to have an efficient way to split payments on-the-fly.
- They usually find themselves splitting bills in the same groups (i.e. same people) multiple times. So, this product should have a functionality to create groups among which payments can be split for longer durations.
- Most interviewees spend long durations of time splitting expenses manually using penpaper or calculators. So, a convenient digital way would help them do this task faster and keep track of the same for later, as many find that in such situations one or two people pay the entire bill and the rest pay back the dues later.
- They don't always split the bill equally; They may split according to each person's share. This means the platform requires a mechanism to enable uneven splits. Moreover, one or two people may pay upfront which is returned to them later.

## Online Survey

We created a google form survey to understand more individualistic properties of people's behaviour when using payment platforms. This survey helped us gain insight into a broader audience's state of mind. The form got 64 responses.

First off, we'd like to get know what age group you fall in?

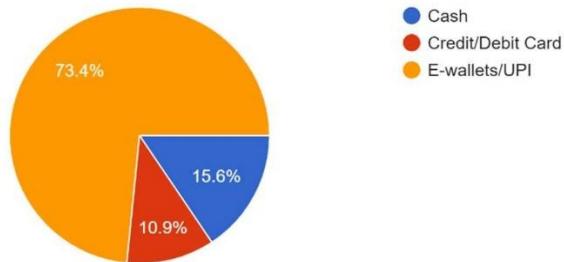
64 responses



We were able to reach out to mainly people in the 18-22 age group. This is the age group that includes people in college. This works for us as our main target audience is people of this age group.

On a day to day basis, which is the most convenient mode of payment for you?

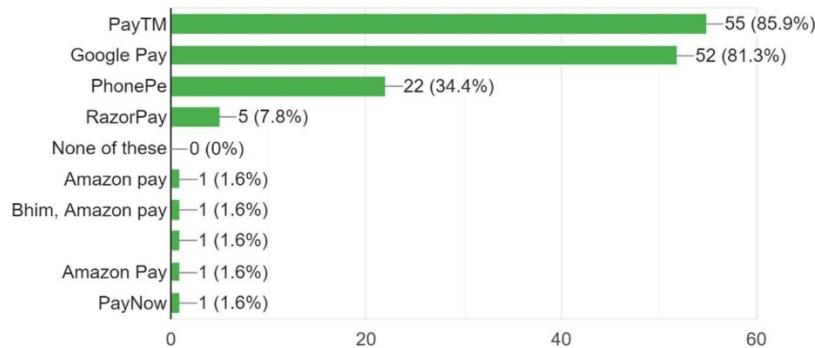
64 responses



This shows us that the majority (73.4%) of the people who took part in the survey, find Ewallets and online payment portals as the most convenient means of payment.

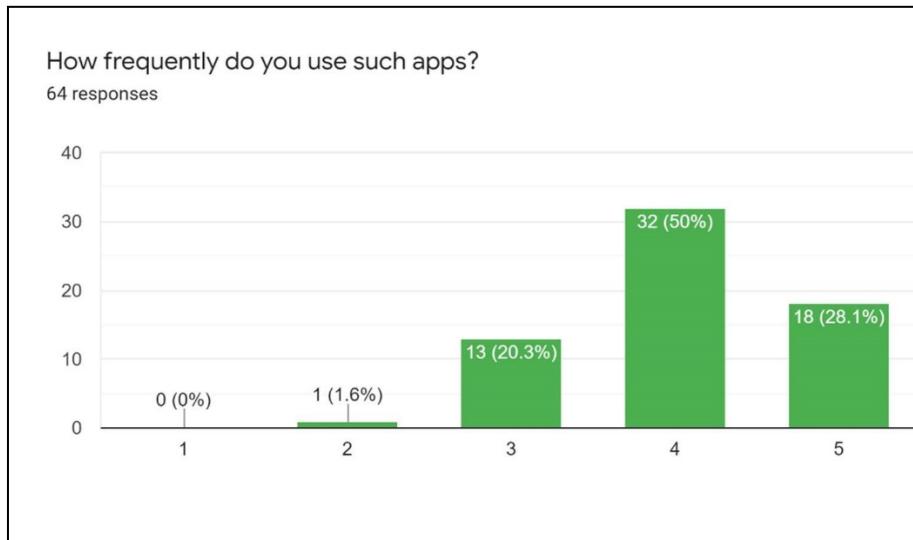
Please select all the Payment platforms that you use/have used

64 responses

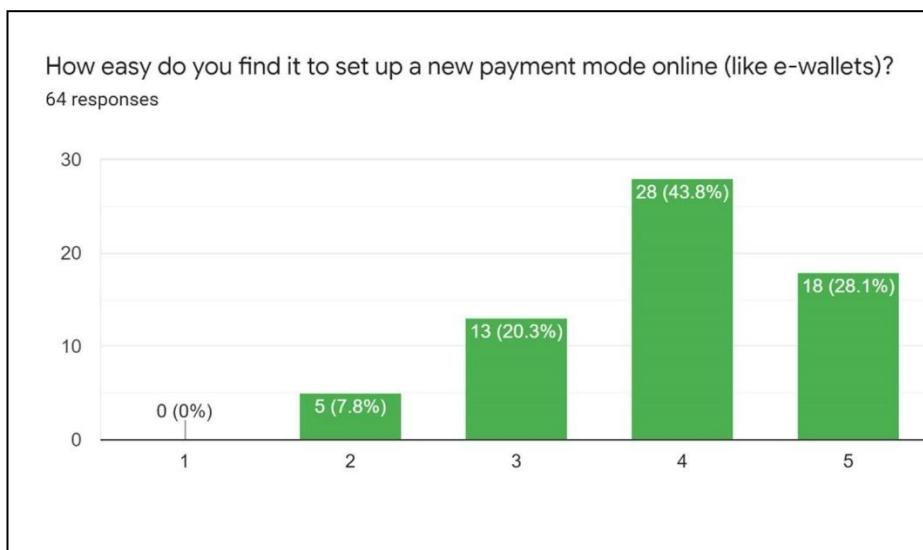


As can be seen from this bar graph, all the people who took part in the survey have experience with

some kind of a payment platform. PayTM and Google Pay are the most used apps in this section of people, while some also use PhonePe.



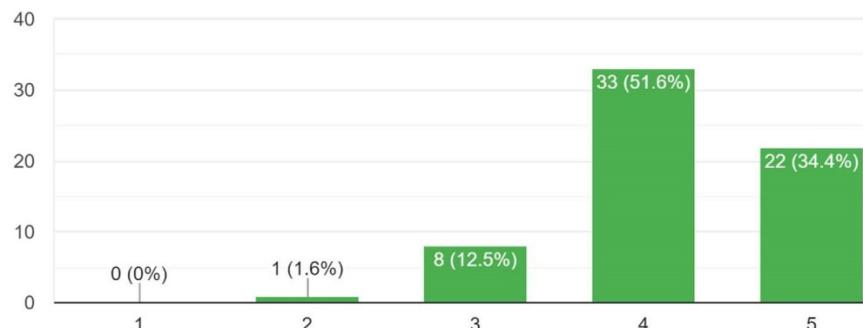
Most people thus use such apps very frequently, out of which 28.1% claim they use such platforms for almost all payments. Therefore, we can say there is huge scope for our project and people would readily be open to using new apps if they have ease of use.



Most people (71.9%) found it easy to set up a new payment mode, whereas the rest (28.1% which is a considerable number) were not necessarily satisfied with the set up modules.

How much would you rate the Ease of Use of the current mobile payment methods?

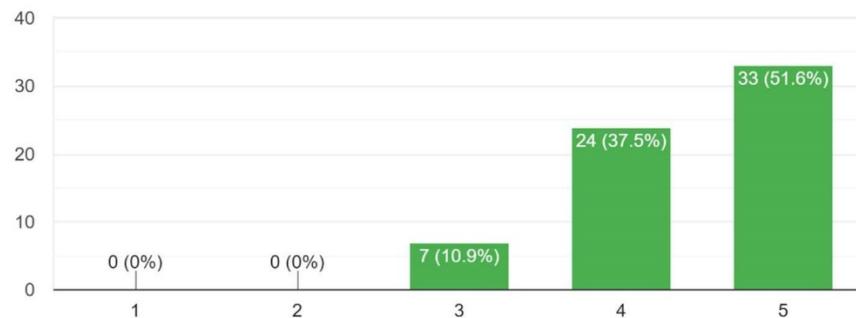
64 responses



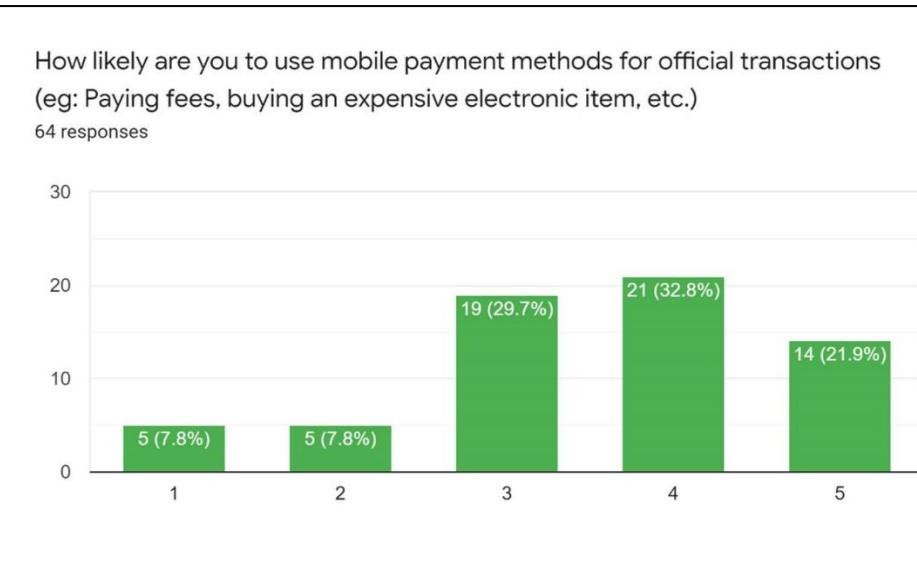
Therefore, large majority of people are happy with the ease of use of their current mobile payment methods. So we can learn a lot about good design practices for such a platform by assessing good examples of online payment methods, like PayTM and Google Pay.

How likely are you to recommend someone to use mobile payment methods?

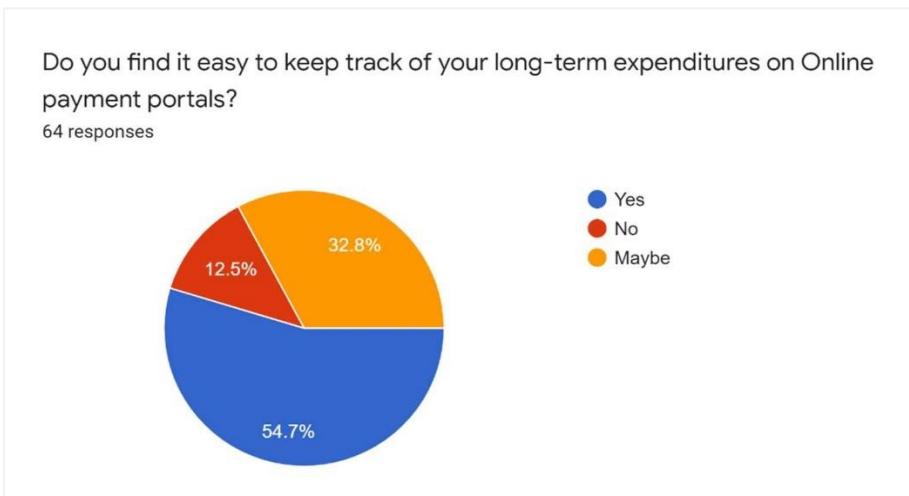
64 responses



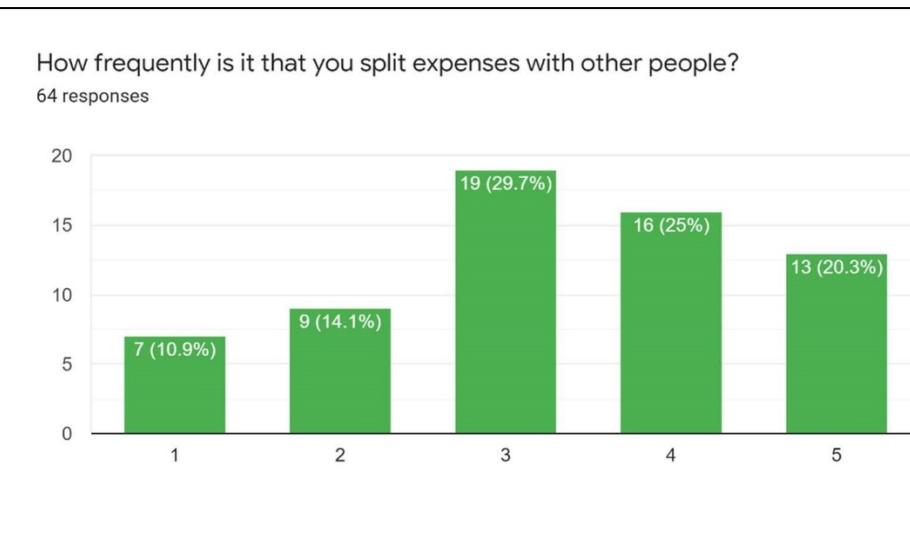
Almost all would strongly recommend the use of such platforms due to their ease of use.



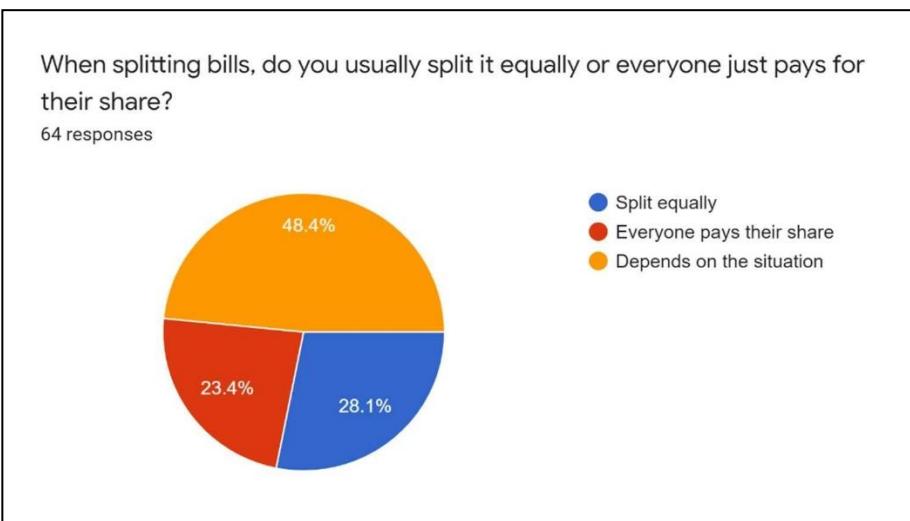
This is one of the arguable parameters of the survey. Though a high portion of the group does trust mobile payments for such big official transactions, a good chunk of them are not entirely sure about trusting such platforms for bigger transactions.



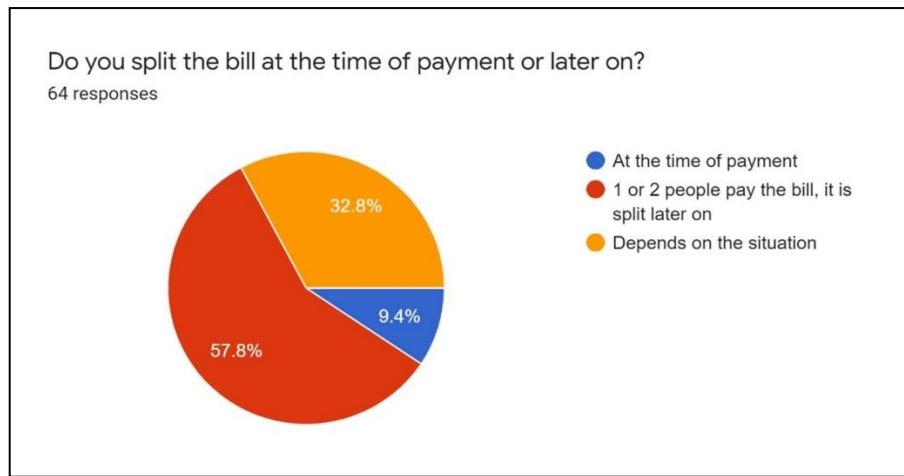
More than half of the people find it easy to keep track of their transactions, but there is a large part of the group which is unsure about the same. So we should ensure that in our product, the "Maybe" turns into a "Yes" and give easier/more direct access to transaction history.



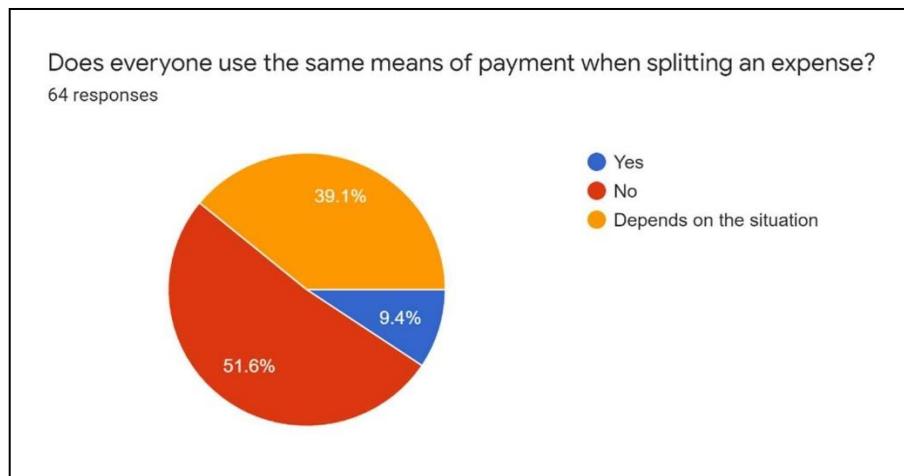
This is again a question with varied answers. Though 25% of the people rarely split expenses, the rest do regularly find themselves in social situations where splitting an expense is required. However, the frequency in this 75% varies largely. We can still conclude that it is a common social activity.



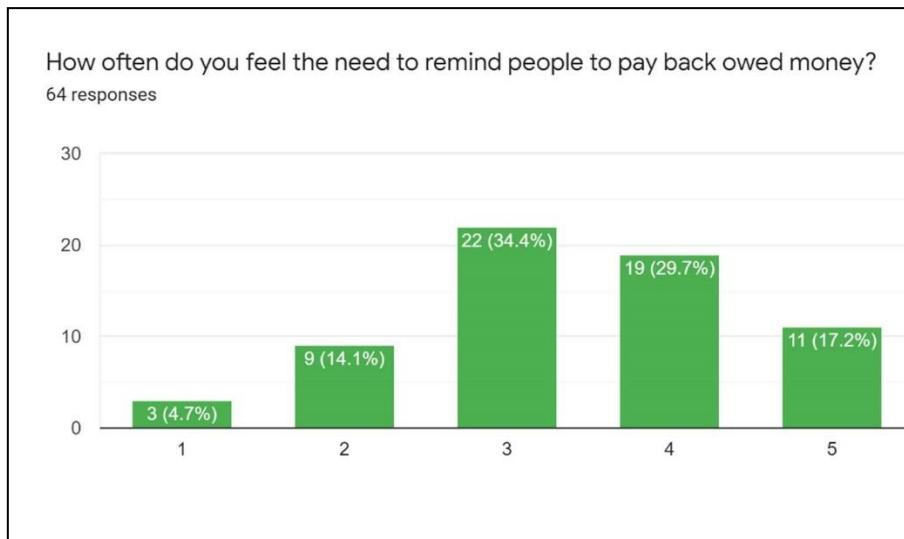
This means each time a bill is split, there is a difference in the way it is split. It may depend on the situation, the people, and the kind of expense of how the people divide the payment. So, our product should be ready to split a payment in any way the user desires.



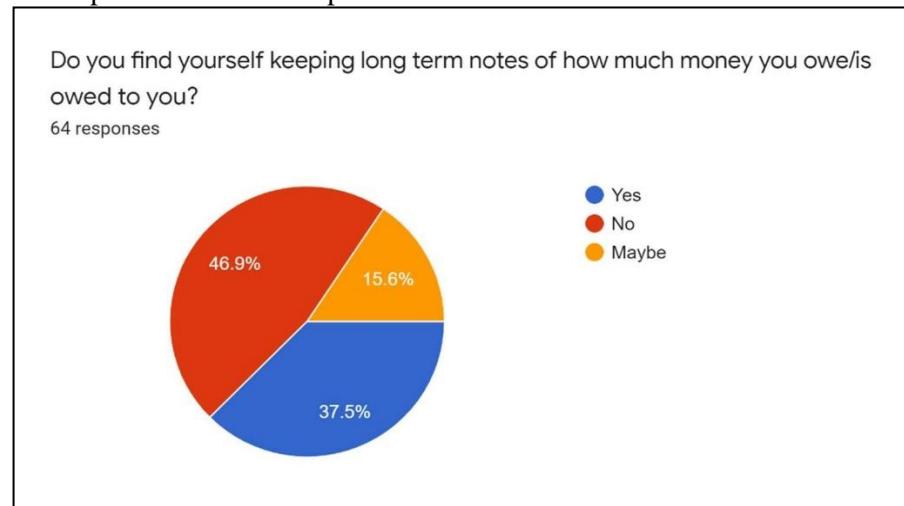
57.8% of the people have recounted that the bill is not split at the time of payment. However, 32.8% say that it depends on certain factors. Hence, again we should make the system be capable of giving the users options to split expenses whenever they like as it varies with the situation.



As most people say there is inconsistency in the mode of payment when splitting expenses, we should make sure our system allows Cash or other methods to act as interleaving methods of payment. This means the user should be able to settle up expenses using cash and then be able to update the same on the system.



This question got varying answers but the deviation is towards higher side. So, an additional functionality would be to let the user remind other users about any standing dues between them. For this, we can provide additional/optional functionalities.



Again, we have gotten a lot of varied responses. So an additional functionality would be good to let the user keep track of their expenses. This can act as a secondary functional requirement.

## 2.1 What We Have Gathered from Requirement Analysis

We have a lot of insight into how our target users behave while making payments, on their own, or in social situations. This can help us in clearly defining our project's goals, scope and requirement:

**Goal:** The goal of this project is to develop a mobile payment platform (an application) which lets its users easily pay for expenses and split those expenses amongst groups of two or more. These splits should occur in whichever way the user demands as there is no fixed way of splitting bills.

**Problems in Existing Systems:** Existing systems include big scale payment apps like PayTM and Google Pay. Their users are largely satisfied with their services. However, all such apps lack the ability to split expenses, which would be unique in our product. Moreover, by learning from the mistakes of smaller scale payment apps we are implementing a user centred approach to understand

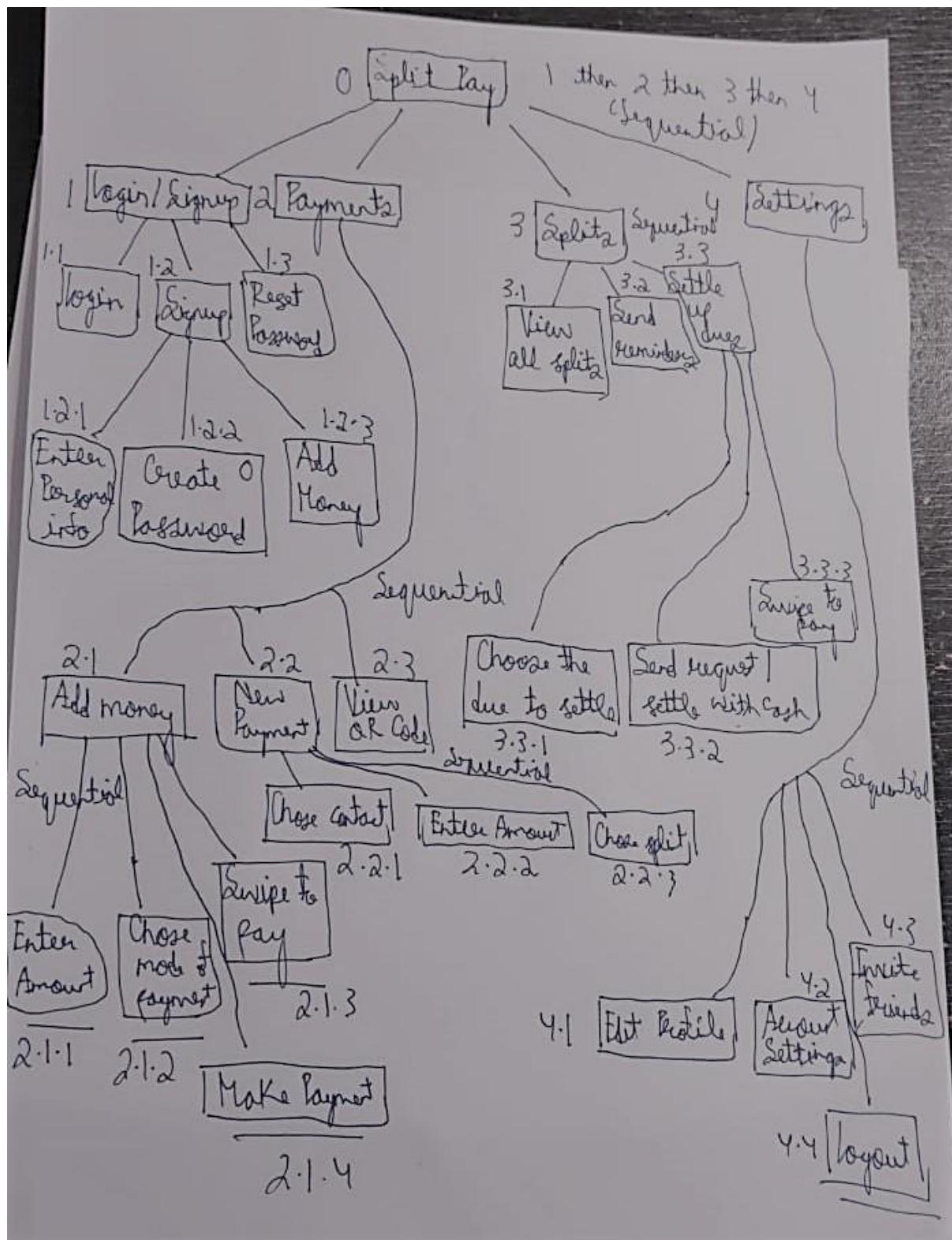
what the users want.

Context of use of the Design: This design will look to aim young adults as its primary users. It is an app which would be used in a social context, when people go out and spend money as a group.

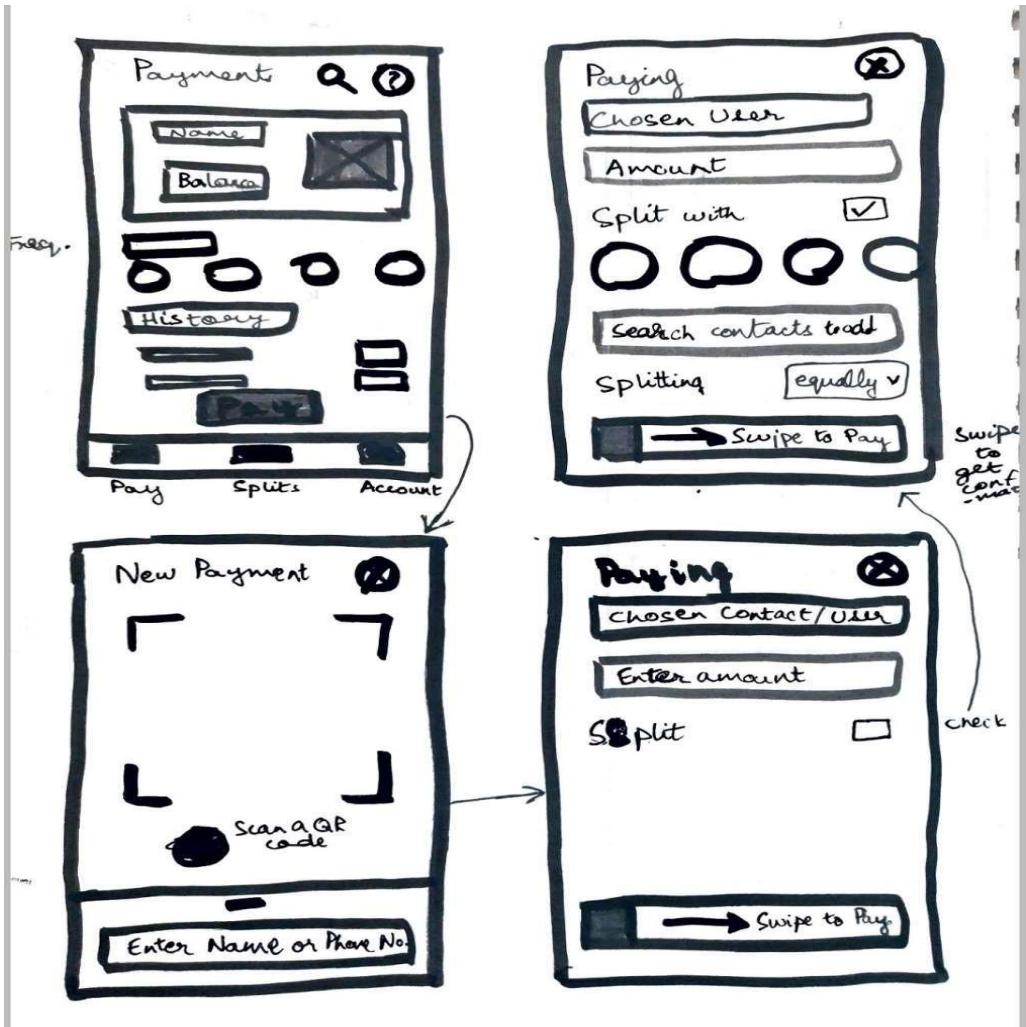
Moreover, due to the Coronavirus situation in 2020, this app also promotes Contactless Payments and thus can be used as a safe means of payment.

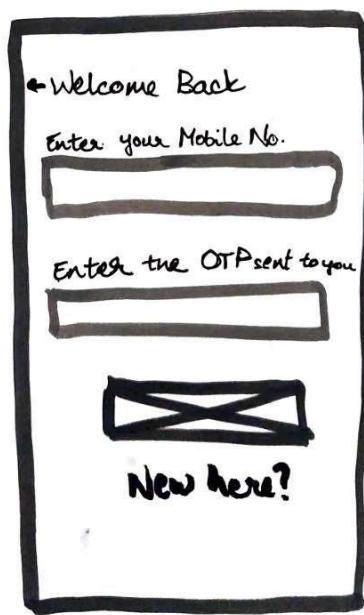
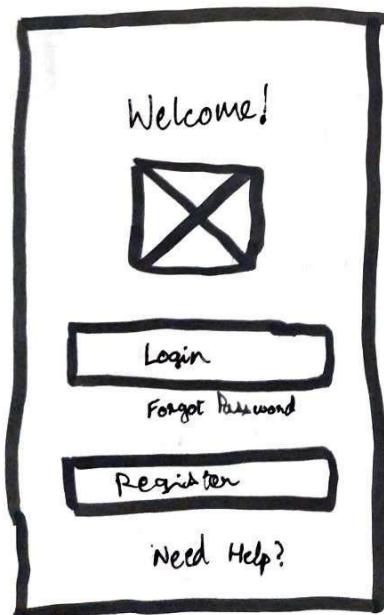
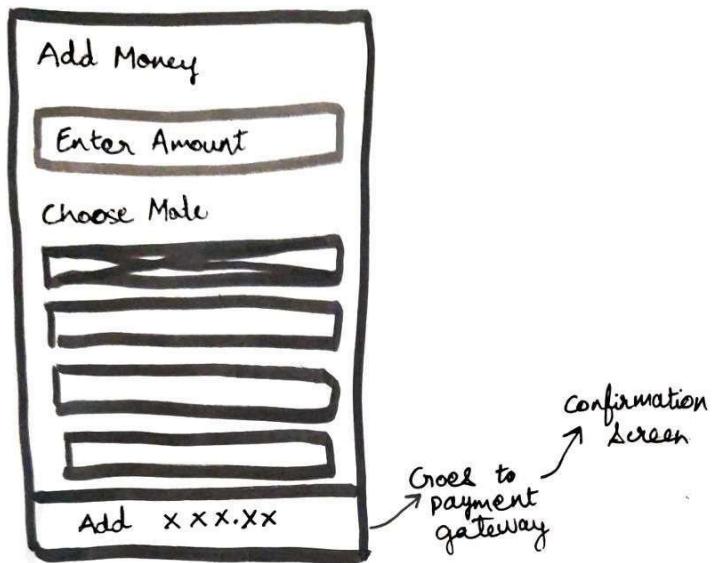
## 4.Data Flow

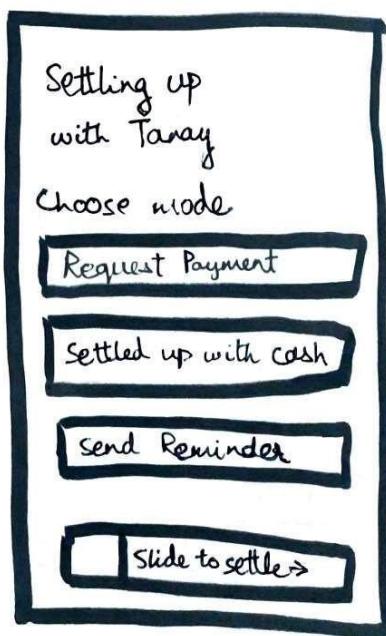
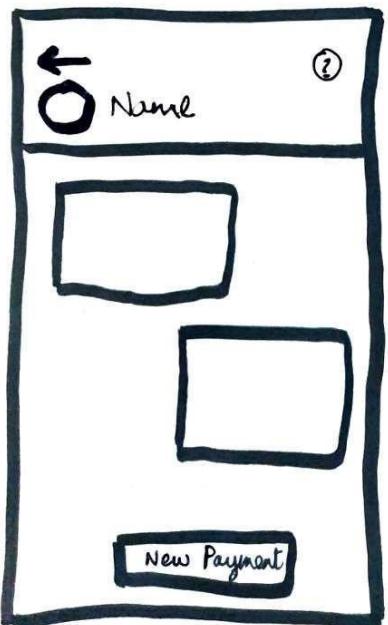
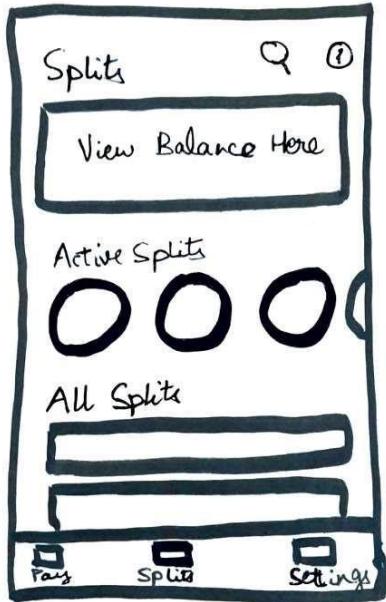
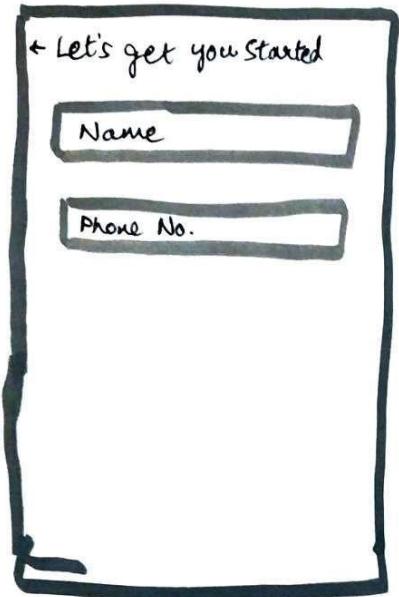
### 4.1 HTA Diagram



## 4.2 Storyboard







## 5. 10 Heuristic evaluation & Shneiderman's 8 Golden Rules matching with UI

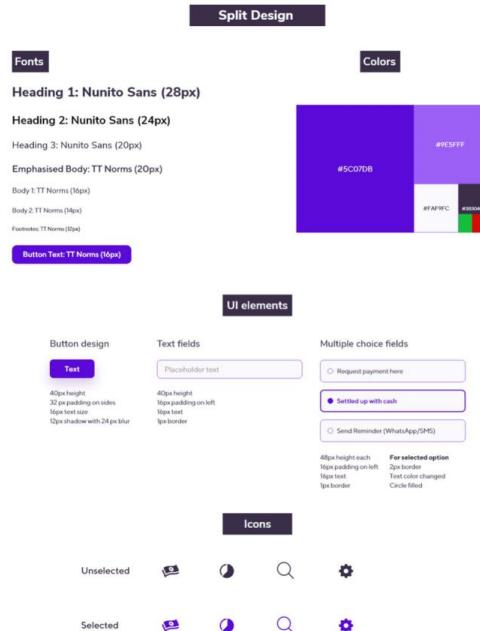
### Evaluation of SplitPay using Schneiderman's 8 Golden Rules

All examples have been marked with a blue box

#### Rule 1: Strive for consistency

To maintain consistency in our project, we came up with a design system called Split Design. Split Design is a collection of fonts, colors, icons and interface elements that help us ensure that our app design is consistent throughout. You can view the various elements of the system as shown:

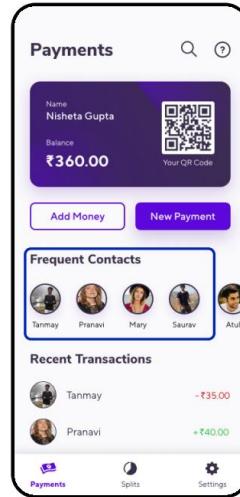
Furthermore, you can view in the screens that very important buttons are always placed at the bottom, to go with Hick's Law. Also, a 8pt grid maintains visual consistency.



The image shows a 'Split Design' UI elements section. It includes a color palette with hex codes (#5C0D8B, #E63FFF, #00AABC, #00AABC, #00AABC), a font specification for 'Heading 1: Nunito Sans (28px)', and a grid-based layout for UI components like buttons, text fields, and dropdowns, all using a 8pt grid system.

#### Rule 2: Cater to Universal Usability (Enable frequent users to use shortcuts)

Our system aims to make the lives of its frequent users by giving them the option to save information like credit card information, and by giving them a list of their frequently interacted contacts.

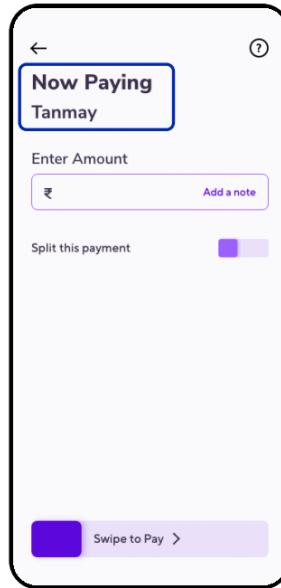
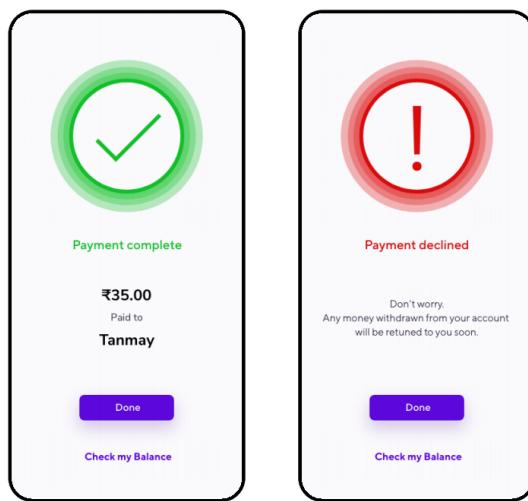


As you can see here, we have provided a list of frequent contacts, with whom the user frequently does transactions/bill splitting with. This helps the user quickly select these contacts and start payments in one click.

### Rule 3: Offer informative feedback

Feedback is extremely important to inform the user the state of the system and in this case, the state of a transaction. That's why ever screen has a big heading telling the current state.

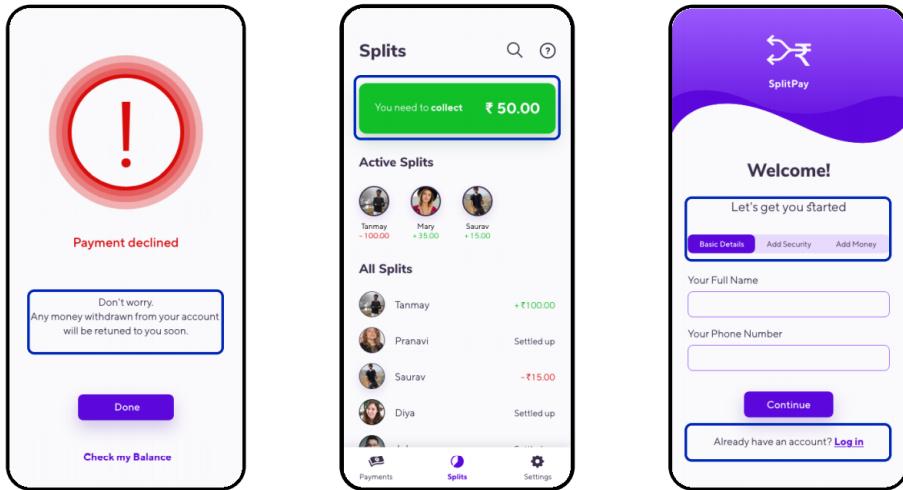
Informative confirmation and alerts on completion or decline of a payment respectively



By giving the name of the contact below "Now Paying" (in this case Tanmay), we alert the user of the contact they have selected for payment.

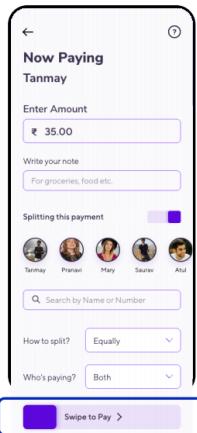
## Rule 4: Design dialogs to yield closure

The language used in the copy text is very simple to understand. It was noted during data collection that our target user would know conversational English and thus the dialogs in the system will be very easy for them to understand. Some examples are:



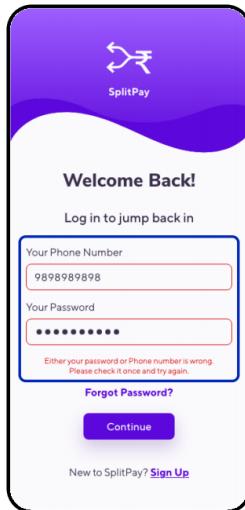
## Rule 5: Offer error prevention and simple error handling

Error prevention has been best handled in the Log in form and the Now Paying pages. On the Now Paying page, we have used a Swipe Mechanism to confirm payment. This is to prevent any slips that may cause the user to select that option. If it was a regular button, a slipped tap would cause a wrong payment to go through, which is not ideal. Meanwhile, in a swiping mechanism, chances of this are less.



## Rule 5: Offer error prevention and simple error handling

In the log-in form, while taking password, the user will be prompted if any details are invalid as shown below:



## Rule 6: Permit easy reversal of actions



As can be seen on all screens of the app, there is always a "Go back" arrow button on the top left corner, to allow the user to go back to the previous page or module.

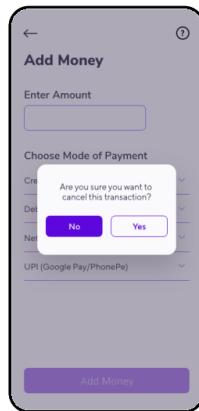
The only place where we haven't put a "Swipe to..." functionality is in the Add money page. This is because, after initiating the transaction, the app will give the user a brief time period to cancel the transaction as shown below

## Rule 7: Support internal locus of control

Letting the user feel in-charge of the entire flow is very important. Therefore, there are certain places and hints where the user is asked for permission to do tasks.

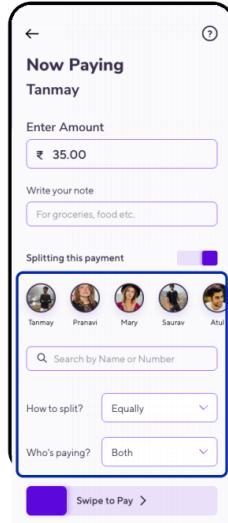
One such example is the Swipe to Pay mechanism. It reassures the user that the payment is only initiated on swiping the bar. The mechanism is shown above.

Another example is when you try to cancel a transaction, you are asked for confirmation. This lets the user feel like they are in control of whether the transaction will go through or not. An example is shown below:

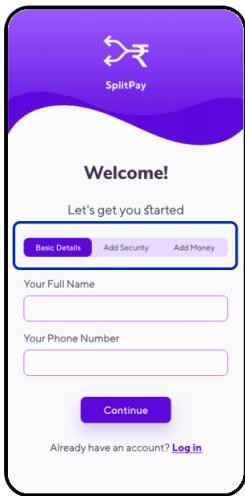


## Rule 7: Support internal locus of control

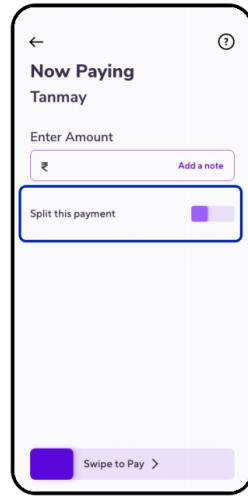
Lastly, a lot of customisability is provided to the user when it comes to splitting expense. For example, you can select who is paying the bill currently and how it is split. This lets the users decide the way the bill is split.



## Rule 8: Reduce short-term memory load



This has been done on the Sign up screens. The entire process is divided into 3 segments so that the user can focus on completing just one task at a time.



Moreover, the expense splitting options are hidden at first, so as to not confuse the user with too many choices. These can be toggled on by clicking on the split this payment button.

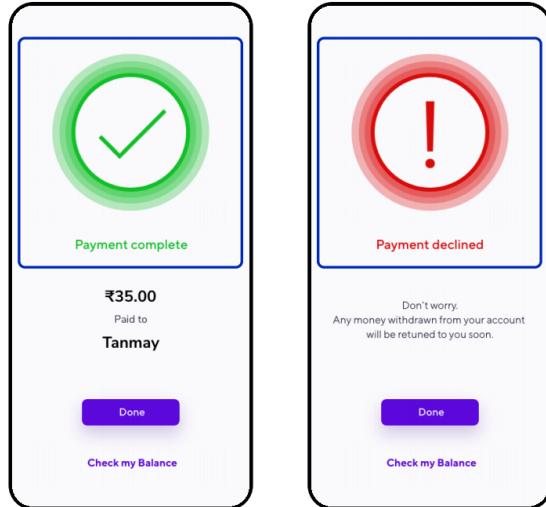
## Evaluation of SplitPay using 10 Usability Heuristics

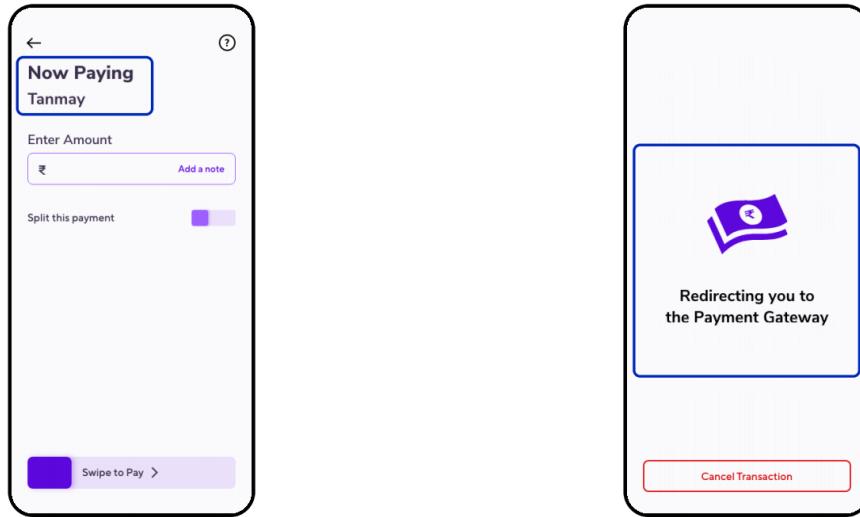
All examples have been marked with a blue box

### #1: Visibility of system status

The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time. When users know the current system status, they learn the outcome of their prior interactions and determine next steps. Predictable interactions create trust in the product as well as the brand.

Informative confirmation and alerts on completion or decline of a payment respectively



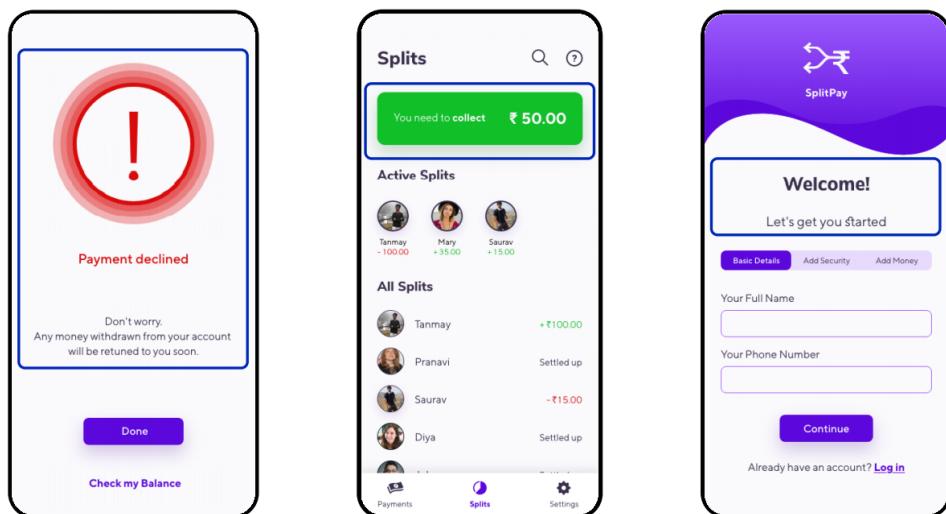


By giving the name of the contact below "Now Paying" (in this case Tanmay), we alert the user of the contact they have selected for payment.

Giving the user proper feedback and showing them what is going on in, and why the app is taking time to "load" the confirmation page

## #2: Match between system and the real world

The language used in the copy text is very simple to understand. It was noted during data collection that our target user would know conversational English and thus the dialogs in the system will be very easy for them to understand. Some examples are:



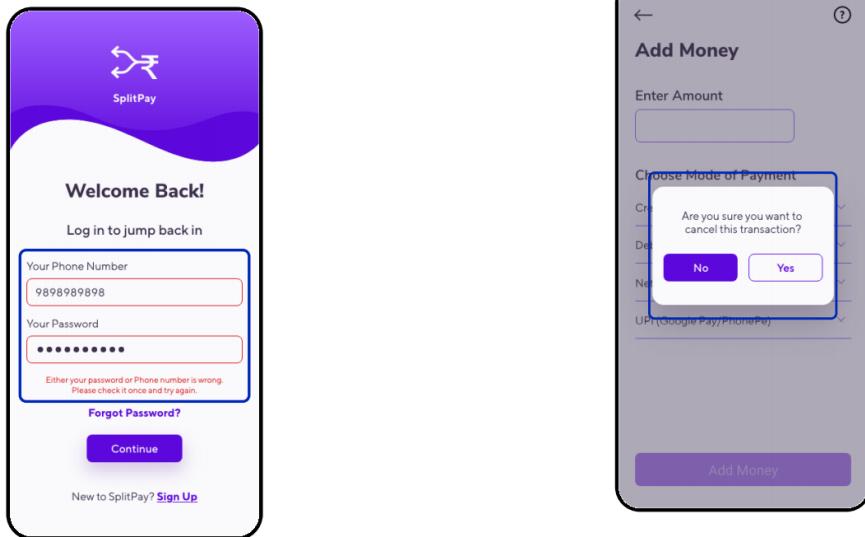
### #3: User control and freedom



As can be seen on all screens of the app, there is always a “Go back” arrow button on the top left corner, to allow the user to go back to the previous page or module.

The only place where we haven't put a “Swipe to...” functionality is in the Add money page. This is because, after initiating the transaction, the app will give the user a brief time period to cancel the transaction as shown below

### #5: Error prevention

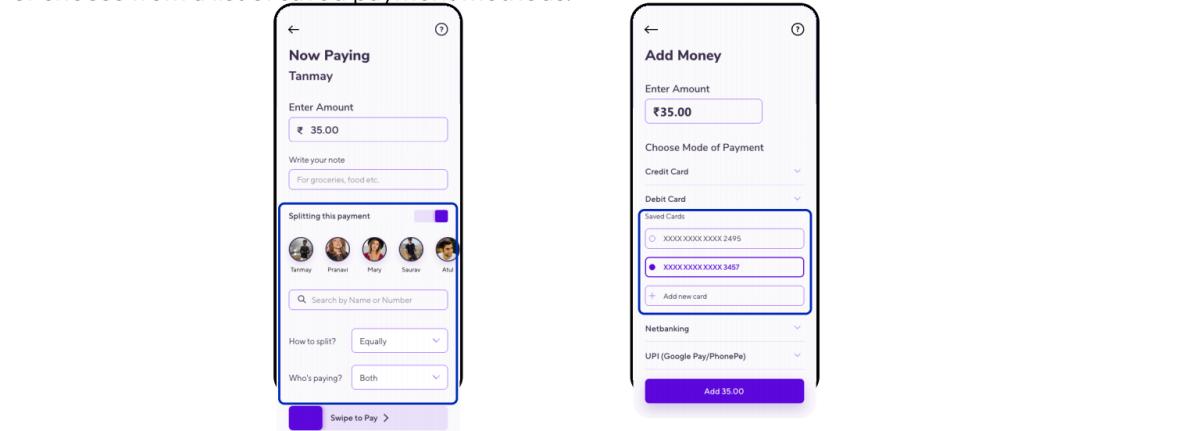


In the log-in form, while taking password, the user will be prompted if any details are invalid as shown

## #6: Recognition rather than recall

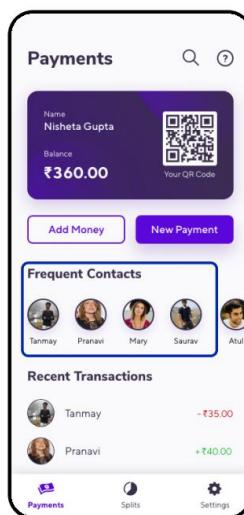
The objective here will be to reduce the memory load the user by making all their options for a task visible. In the Payment Module, while determining the splitting parties, the user can choose to type each name (RECALL) or select from a scrollable list of all available users within the group (RECOGNIZE).

Also, within the Payment Module, the user can choose to add the details for their payment method or choose from a list of saved payment methods.

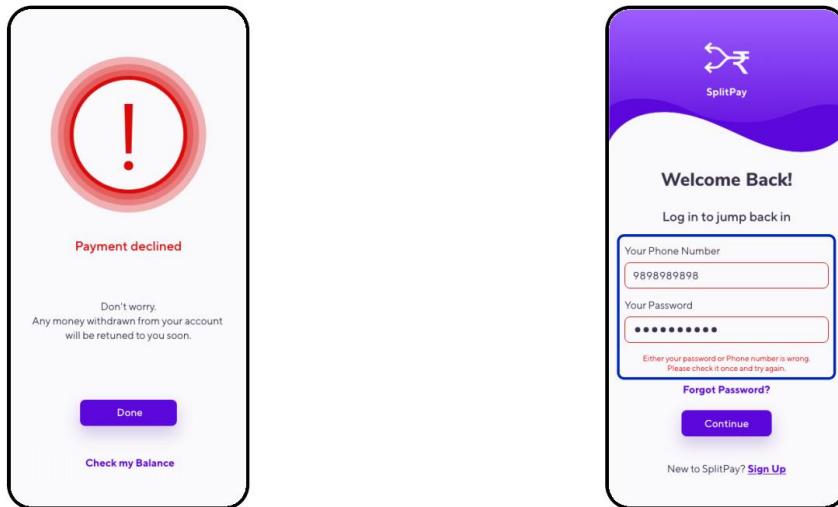


## #7: Flexibility and efficiency of use

An active user can view their frequent contacts. This will make the app more efficient for them and also be easier and faster to use.



## #8: Help users recognize, diagnose, and recover from errors



Informative alerts on decline of a payment

In the log-in form, while taking password, the user will be prompted if any details are invalid as shown

## #10: Aesthetic And Minimalistic Design

A minimalism is not only a fashion of last few years, but it certainly is a lasting trend with the aim to reduce the description of a subject just to its necessary elements. It has many applications in art, music, and literature. Minimalism helps users to quickly access important information and come to the result quickly.

## 6.1. Unit Testing

Registration/Sign Up Page:

### 1. Testing Name

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering name with a digit	Atul123	Proceed to successfully register on the site.	Registration failed, as no digits allowed in name	FAIL
b	Entering valid name	Atul	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS

### 2. Testing Phone Number

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering more/less than 10 digits	88797285989 887972859	Proceed to successfully register on the site.	Registration failed, only 10 digit number allowed	FAIL
b	Entering valid name	8879728598	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS

3. Testing Email:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering invalid email format	Atul.com	Proceed to successfully register on the site.	Registration failed, only valid email formats allowed	FAIL
b	Entering valid name	atul@gmail.com	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS

4. Entering Password

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering password with no digits	atulagarwal	Proceed to successfully register on the site.	Registration failed, password needs digits	FAIL
b	Entering password of length less than 8	Atul123	Proceed to successfully register on the site.	Registration failed, password needs at least 8 characters	FAIL

c	Entering Valid Password	Atulagarwal123	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS
---	-------------------------	----------------	---	---	------

**5. Confirm Password Test:**

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering a password that does not match	Password: Atulagarwal123  Confirm Password: atulagarwal	Proceed to successfully register on the site.	Registration failed, fields must match	FAIL
b	Entering matching Passwords	Password: Atulagarwal123  Confirm Password: Atulagarwal123	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS

**6. Adding an Image**

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Uploading a pdf/ ppt/ any other file format	Da.doc/da.pdf	Proceed to successfully register on the site.	Registration failed, only image type files allowed	FAIL

b	Uploading a file of image type	Pic.png	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS
---	--------------------------------	---------	---	---	------

#### 7. Testing Repetitive Entry:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering an email/phone number that exists already	atul@gmail.com or 8879728598	Proceed to successfully register on the site.	Registration failed, user already exists	FAIL
b	Entering completely new values	agarwal@gmail.com	Proceed to successfully register on the site.	Proceed to successfully register on the site.	PASS

**Testing Login Page:**

**8.Entering Credentials:**

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering an invalid/unrecognized phone number	88797285989/ 1234657890	Proceed to successfully log on the site.	Login failed, invalid format or not recognized error shown	FAIL
b	Entering valid and identified phone numbers	8879728598	Proceed to successfully log on the site.	Proceed to successfully log on the site.	PASS

**9.Testing Password Credential:**

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering an invalid/unrecognized password	Password: Atulagarwal123  Confirm Password: Atulagarwal12	Proceed to successfully Login the site.	Login failed, password incorrect must match	FAIL
b	Entering correct Passwords	Password: Atulagarwal123  Confirm Password: Atulagarwal123	Proceed to successfully Login on the site.	Proceed to successfully Login on the site.	PASS

10.Add new Split:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Entering a negative number as Amount	-200	Proceed to add split to the ledger	Failed, only positive amounts can be splitted	FAIL
b	Entering 0 as split amount	0	Proceed to add split to the ledger	Failed, as amount must be greater than 0	FAIL
c	Entering a positive non zero amount	200	Proceed to add split to the ledger	Proceed to add split to the ledger	PASS

11.Adding Note:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Leaving the note empty	""	Proceed to add split to the ledger	Failed, note must have a value.	FAIL
b	Entering a note	“Pizza”	Proceed to add split to the ledger	Proceed to add split to the ledger	PASS

12.Adding Numbers to Split Payment with:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Adding invalid number format	88797285989	Proceed to add split to the ledger	Failed, phone number must be 10 digit long only.	FAIL
b	Adding a number that does not exist on platform	1234568790	Proceed to add split to the ledger	Failed, number not recognized	FAIL
c	Adding own number	8879728598	Proceed to add split to the ledger	Failed, number cannot be your own	FAIL
d	Adding valid numbers	8879728599	Proceed to add split to the ledger	Proceed to add split to the ledger	PASS

13.Trying to Pay:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Trying to add payment without any number	""	Proceed to add split to the ledger	Failed, need to click on add number button first.	FAIL
b	Adding payment after clicking add number button	8879728598	Proceed to add split to the ledger	Proceed to add split to the ledger	PASS

14.View Recent/ All transactions on the main page:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Failure to fetch from the API	Error on client machine/ server status	Proceed to view transactions	Failed nothing fetched.	FAIL
b	Successful fetch	Correct fetch link with key	Proceed to view transactions	Proceed to view transactions	PASS

15.View Settle Page:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Make Transition to Home/Setting Page	Href access to user respective page	Proceed to respective Page	Page Loads	PASS
b	Choose to settle active transaction	Choose correct settle contact page	Proceed to view history	Specific transaction page loads	PASS

16.Settle Payments:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	View history with specific user	Select correct user detail.	Proceed to View correct history	All user history in table format displayed	PASS
b	Make a payment to the user, to settle payment.	Choose settle option	Proceed to payment page	Page correctly identifies both parties	PASS

17. Settle Options:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Settle With Cash	User balance both users and ID.	Book settled	Balance between both users settled.	PASS
b	Settle with UPI/Online Payment	User online card details.	Book Settled	Works only in testing phase, cannot make transaction (key not acquired)	FAIL

18. OR Code Development:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Make user specific QR code.	User phone number on weak network.	Generate, user QR code.	QR Code generated after long delay.	PASS
b	Make efficient QR code.	User phone number on weak network.	Generate, user QR code.	QR code generated in time.	PASS

19.QR Code scan:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Scan QR Code using UPI app.	Generated QR Code.	User payment details.	Key required, cannot generate at this point.	FAIL
b	Scan QR Code using QR code scanner.	Generated QR Code.	User payment details.	User payment details.	PASS

20.User Help:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Request Help Document	Request HREF access.	User help documentation page.	User help documentation page.	PASS

21. User Navigation

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Click on Navigate Link	Request HREF access.	Specific page loaded	Respective page access given	PASS

22. User Switch Between Sign Up and Login:

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Click on link for page	Request HREF access.	Specific page loaded	Respective page access given	PASS

23. Successful Fetch from API

Test No	Function	Input	Expected Result	Actual Result	Pass/Fail
a	Request Data using token and Display	Fetch on page load	Correct data display	Data fetched and displayed	PASS

## 6.2 Usability Testing:

The usability evaluation of the SplitPay was conducted by users.

During the usability evaluation, 4 participants, were asked to play around the site. During this time, participants:

- § Completed a user background questionnaire
- § Answered questions about initial site impressions
- § Performed real-world tasks on the site while thinking aloud

Anchit Mishra	19BCE2578
Aditya Poonia	19BCE2251
Jatin Sohlot	19BCE2256
Palak Teotia	19BCB0126

No.	Test Description	Test Status
1.	Land on First Page, proper load	User 1: PASS User 2: PASS User 3: PASS
2.	Enter Proper Details to Sign Up	User 1: PASS User 2: PASS User 3: PASS
3.	Directed to Login Page	User 1: PASS User 2: PASS User 3: PASS
4.	Enter Correct Credentials to Login	User 1: PASS User 2: PASS User 3: PASS
5.	Directed to Home Page	User 1: PASS User 2: PASS User 3: PASS
6.	Click on New Payment to User	User 1: PASS User 2: PASS User 3: PASS

7.	Enter user Phone Number, Note and Amount	User 1: PASS User 2: PASS User 3: PASS
8.	Click to Add Payment	User 1: PASS User 2: PASS User 3: PASS
9.	Details saved and directed to home	User 1: PASS User 2: PASS User 3: PASS
10.	Add New Split	User 1: PASS User 2: PASS User 3: PASS
11.	Enter all details and click add	User 1: PASS User 2: PASS User 3: PASS
12.	Details saved and directed to Home Page	User 1: PASS User 2: PASS User 3: PASS
13.	Go to Settle Page and Click on Settle Button	User 1: PASS User 2: PASS User 3: PASS

14.	Transaction history with user loaded choose to settle.	User 1: PASS User 2: PASS User 3: PASS
15.	Choose Cash as settle option.	User 1: PASS User 2: PASS User 3: PASS
16.	Choose Online Payment as payment option.	User 1: PASS User 2: PASS User 3: PASS
17.	With Cash – Split Settled go to splits page	User 1: PASS User 2: PASS User 3: PASS
18.	With online – Split Settled go to splits page.	User 1: FAIL User 2: FAIL User 3: FAIL
19.	Go to user setting page, load QR code	User 1: PASS User 2: PASS User 3: PASS
20.	User Logout	User 1: PASS User 2: PASS User 3: PASS

Total Cases: 20

Total Pass: 19



Question	Name	Response
How was your overall experience?	Achint Arya Mishra	It was excellent! I found it to be very useful
	Jatin	It was good
	Palak Teotia	It was pretty nice
Did you face any problems in any of the units?	Achint Arya Mishra	No, I could easily navigate throughout the interface
	Jatin	Not really, everything was clear

	Palak Teotia	I got a little stuck at the splitting module but it was all clear after looking at the tutorial in the app
Do you have any suggestions for us?	Achint Arya Mishra	A dark mode of the app would look really good
	Jatin	No not really
	Palak Teotia	Nothing
Did you see any shortcomings in the site?	Achint Arya Mishra	No
	Jatin	No
	Palak Teotia	No
How would you rate your overall experience?	Achint Arya Mishra	10
	Jatin	10
	Palak Teotia	10

## 7. Implementation

Code Source: <https://github.com/atulragarwal/SplitPay-HCI>

```
<!DOCTYPE html>

<html>

<head>

<title>SplitPay</title>

<link rel="preconnect" href="https://fonts.gstatic.com">

<link rel="stylesheet" href="../CSS/login.css">

<link rel="icon" type="image" href="../Images/icon.jpg">

<link rel="stylesheet" href="../CSS/payments.css">

<link rel="stylesheet" href="../CSS/pay.css">

<link href="https://fonts.googleapis.com/css2?family=Nunito+Sans&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<!-- <meta name="viewport" content="width=device-width, initial-scale=1.0" -->

<script src="https://kit.fontawesome.com/bc6f27e589.js" crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

</head>

<body>

<div class="m-container">

<div class="b-head">
```

```
<a href="home.html"><i class="fas fa-arrow-left"></i></a><a style="float: right;" href="https://docs.google.com/document/d/1XRGH7iECFYXqqP-rYERKpPSdEJZUMrsEKaR2Om5JhM0/edit?usp=sharing"><i class="far fa-question-circle"></i></a><br>

<h3>Now Paying</h3>

</div>

<div class="paymentForm">

    <form onsubmit="return createPayment()" action="home.html"
id="paymentDetailsForm">

        <label for="payeeNumber">Enter Phone Number:</label><br>

        <input class = "amtInput" id="payeeNumber" type="tel" required
placeholder="XXXXXX XXXXXX"><br><br>

        <label for="paymentAmt">Enter Amount</label><br>

        <input class="amtInput" id="paymentAmt" type="number" required
placeholder="Rs.XX.XX"><br><br>

        <label for="paymentNote">Enter Note</label><br>

        <input class="amtInput" id="paymentNote" type="text" required
placeholder="Note"><br><br>

        <input value="Pay" type="submit">

    </form>

</div>

<div class="bottomNav" style="background-color: rgb(242,242,242);>

    <div class="paymentOption">

        <i class="fas fa-money-bill-wave-alt"></i>

        <p><a href="home.html">Payments</a></p>

    </div>
```

```
<div class="paymentOption">

    <i class="fas fa-chart-pie"></i>

    <p><a href="splitCollect.html">Splits</a></p>

</div>

<div class="paymentOption">

    <i class="fas fa-cog"></i>

    <p><a href="userSettings.html">Settings</a></p>

</div>

</div>

</script>

let tokenId = sessionStorage.getItem('userToken');

console.log('Token '+tokenId);

document.getElementById('paymentDetailsForm').addEventListener('submit', (e) =>{
    e.preventDefault();
})

function createPayment(){

    document.getElementById('paymentDetailsForm').addEventListener('submit', (e) =>{
        e.preventDefault();
    });
}
```

```
let statusCheck = 'true';

let payAmt = document.getElementById("paymentAmt").value;

let payReceiver = document.getElementById('payeeNumber').value;

let payNote = document.getElementById('paymentNote').value;

if(payReceiver.length != 10){

    alert("invalid phone number format.");

    return false;

}

fetch('http://127.0.0.1:8000/splitpay/payment/{

    // mode:'no-cors',

    method: 'POST',

    headers: {

        'Content-Type': 'application/json',

        'User-Agent': 'PostmanRuntime/7.28.0',

        'Authorization': 'Token ' + tokenId

    },

    body: JSON.stringify({

        "amount": parseInt(payAmt),

        "reason": payNote,

        "payee": payReceiver

    })

}).then(res =>{
```

```
let resStatus = res.status;

if((resStatus==200) || (resStatus==201)){
    return res.json();
}

else{
    statusCheck = 'false';

    return res.json();
}

})

.then(data => {

    if(statusCheck === 'false'){
        alert(data.message);

        return false;
    }

    window.location.href = 'http://127.0.0.1:5500/Frontend/HTML/home.html';
    console.log(data);})

.catch(error => {
    console.log('error'+error);
})

</script>

</body>

</html>
```

## **Conclusion:**

Through this project we have concluded that a payments maintenance app including the SplitPay functionality is a great feature which has a lot of good responses from the survey conducted by us. Also we concluded through testing that this project is ready for deployment once API access code from Google Developers portal is received.