

2020



Splitsy

IS52018C/S: Software Projects

SPLITSY, BILL SPLIT APPLICATION

BISHAI RAI, DAVID CARDOSO, MARY SOUSA, YASMIN PAKSOY

Abstract

Online payment applications are very useful as people can make payments in an efficient and systematic way. In a world more and more focused on privacy we have noticed a lack of bill splitting applications privacy orientated. The purpose of the project was to plan, analyse and gather information to build a secure and user-friendly bill-splitting application. The proposal also focuses on design as well, with the use of techniques that would improve the overall system security and reliability.

We have explored the existing bill splitting applications, their features, weaknesses, and possible solutions. Additionally, we researched about demographics and the different types of data related to the subject. Important design methods such as UML and use cases were also analysed along several relevant tools and technologies needed for the development of this app. Online surveys were conducted on different stages in order to gather vital information and timed tests to measure app's navigation, accessibility, and functionality.

The results revealed that most of the participants approved the layout, colour scheme, navigation, functionality, and importance of the application. Through the numerous feedbacks received, we performed changes on the prototype. Moreover, based on the final online survey, we confirmed that our changes improved the application and users valued the applications user friendliness and reliability.

All the research, surveys and results have proved the importance and need of building a privacy focused and user-friendly application to enhance user experience.

Contact Details

Bishal Rai - brai001@gold.ac.uk

David Cardoso - dcard001@gold.ac.uk

Mary Sousa - msous001@gold.ac.uk

Yasmin Paksoy - ypaks001@gold.ac.uk

Contents

Abstract.....	1
Contact Details	1
Concept	4
Stakeholders	4
Prior Knowledge & Market Survey	4
Prior Knowledge	4
Market Research.....	5
Initial Market Survey.....	5
Secondary Market Survey	6
Design.....	9
Normalization	9
UML Diagram	10
Sequence Diagram.....	11
Activity Diagrams	13
Use-Case Diagrams.....	15
User Journey Mapping Diagram.....	16
Technical & Functional Specification	17
Purpose	17
Scope	17
System Overview	17
Payment API	18
Camera API	18
Functional requirements	19
Non-Functional requirements.....	20
Development Methodology.....	21
Impactful Supervisor Meetings.....	21
Meeting 2	21
Meeting 5	22
Meeting 6	23
Prototyping	23
Timed Tests.....	29
Survey Tests.....	31
Accessibility.....	32
Fonts	32

Colour Contrast	33
Language	34
Button Sizes.....	34
Project Management	35
Ethical Audit	39
Evaluation Plan	39
Performance Evaluation.....	41
Communication	42
Productivity	43
Time management.....	44
Conclusion	46
Glossary	48
References	50
Bibliography	55
Appendices.....	58

Concept

The concept of this application is to facilitate bill splitting between groups while maintaining a sense of privacy. As technology has developed individuals have become more opposed to sharing personal information with others, bank accounts included. This application will allow users to split bills without exchanging bank account information.

This application will allow users to split bills in more sophisticated ways; instead of having to spend time manually calculating each user's share, groups will be able to scan their shared bill and then individually select which items to pay for simultaneously. Additionally, users can also decide to split their bill equally.

When creating their account, users will be asked to add their bank account information to the app, allowing them to send and receive payments from other users. It is necessary that this step is completed at account creation as this application will also ensure security for the person who is owed money; when creating the bill this user will be able to select a pay-by-date which will ensure they have been paid back by that date by automatically taking payment from their account if they have not yet paid back what they owe.

Stakeholders

The main stakeholders for this application are individuals over the age of 18 who want to share a single payment cost with others. Of this group, our main focus includes individuals who are often going out with friends and want an easier way to split and track group payments; our research showed that 18 to 30-year-olds go out more much more often compared to other age groups, so they are our target audience. Our additional focal points include groups who live together and need an easier way to split living costs and individuals who want to make payments without sharing personal data such as their bank account.

Other stakeholders are bank entities, who will be involved in payment transactions, and services where this application will be used at, such as shops and restaurants. This application will make it easier for these services to accept payments as they will not have to accept multiple card payments or give customers separate checks. We, as developers, are also stakeholders in this application as we are responsible for its execution and have a direct interest in it being successful. Our module leader, Sean McGrath, is also a stakeholder as he provided the conditions this project must meet, is the one guiding us through the development of this application and will be the one grading it.

Prior Knowledge & Market Survey

Prior Knowledge

When deciding on the concept for this application we knew bill splitting applications existed but felt that there was a gap in the market; we wanted to provide a way for groups to share bills without having to share any personal information about themselves, which was something that we knew to be impossible.

Market Research

Research into the market showed that this was in fact the case; even our biggest competitor, Splitwise, shared users' personal information, such as their phone number and email address, with others [1]. Other payments apps, such as PayPal, also require a certain degree of personal information to be shared when making payments [2]. This allowed us to make the decision to base this application around privacy.

To decide what type of features we should implement, we did research on existing apps. We found that most existing apps, such as Tab [3] and Splitwise, had features to allow users to scan their bill using their camera, although for Splitwise this feature is only available in the pro version of the application [4]. We decided to follow convention and implement this feature also, making our application more competitive compared to Splitwise, which is the most used bill splitting application with over 105,000 downloads on the Google Play Store [5], as we would provide this feature for free.

During our research into existing apps, we found that users' most common criticism was that they were too complicated; the applications were not streamlined enough so it took too many clicks to get to the desired outcome [5]. Users also found that it was too difficult to edit the list of items produced by the camera scan [6]. Due to this, we made usability one of our non-functional requirements, as we wanted to make sure we did not have the same problems as our competitors, therefore being more competitive.

Users commented that it was complicated to settle payment when a member settles too early as the whole process had to be performed manually again. This helped us to decide that every member must join the bill and only then is payment performed.

Users also commented that their group members often forgot to pay them back even with the use of a bill splitting app. This prompted us to include the automatic payback feature, as otherwise, our application becomes unnecessary; if bill members are not paying the creator back promptly users will no longer see the value in our app.

To decide on our target audience, we did research into which type of groups split bills most often. We found that bills are split the most in restaurant settings and that individuals under the age of 35 spent the most money eating out [7], making them our target audience.

Initial Market Survey

We conducted a market survey to help identify potential new customers and better understand our existing customers. We wanted to know how individuals currently split bills, and which features they would be most interested in so we could better tailor the application towards their needs.

We shared this survey online with friends and on social media platforms. Ideally, we would have been able to survey a diverse group of people by conducting this

survey with members of the public instead of online, however, due to the COVID-19 pandemic we had to adjust our methodologies.

Question	Answer	Occurrence Percentage
Which age group are you in?	18 - 24	57.9
Before COVID-19, how many times per month would you split a bill with others?	Rarely	52.6
How do you split bills?	Bank transfer	52.6
Would you use an app that facilitates splitting a bill among different people?	Yes and Maybe	36.8 each
In a cost splitting app, please classify from 0 to 4 how important would be the following features:		
1. Splitting the bill equally by the number of participants	3	42.1
2. Choosing exactly what item you have consumed	4	52.6
3. Data encryption	4	36.8
4. Privacy	4	36.8
5. Spending report	3	47.3

Figure 1: Most common answers to initial market survey

From the table on Figure 1, we could deduce that our average user would have between 18 and 24 years old; use bank transfer to split bill and is inclined to try a splitting application. To this user, the main features of such application should be privacy, data encryption and being able to choose exactly the consumed item.

During the execution of this survey, we became aware of the fact that some questions did not represent anything distinctive. For example, the options for the first question, "*Which age group are you in?*", did not represent distinguishable groups. We had also not included any layout-oriented questions in this survey, which we later realised would be necessary for the development of our high-fidelity prototype. Due to these factors, we conducted a secondary market survey.

Secondary Market Survey

This survey, much like the first, was conducted online. When creating this survey, we made certain that each question and its potential answers clearly illustrated a certain aspect. For example, for the question "*Which age group are you in?*" the response options represented different age groups, as we had previously found statistics regarding how regularly these age groups ate in restaurants.

For questions measuring an individual's attitude towards certain features, we used a Likert-type scale. We considered using a slider scale but found that Likert scales would be easier for our participants to use, as they would not have to read and understand any labels which could make it "*harder for them to orientate themselves*" (Dobronte, 2017) [8] when answering questions. Preston found that reliability was significantly higher for scales with more response categories and that users much preferred a 10-point scale as they were able to express their feeling more adequately [9], hence we used a 10-point scale in this survey.

From the results of this survey, we found that most individuals split bills 1 to 4 times a month and that 78% of participants split bills in social settings. 66% of participants indicated that they currently split bills using bank transfers and most

participants claimed they would be interested in an application that facilitates splitting a bill.

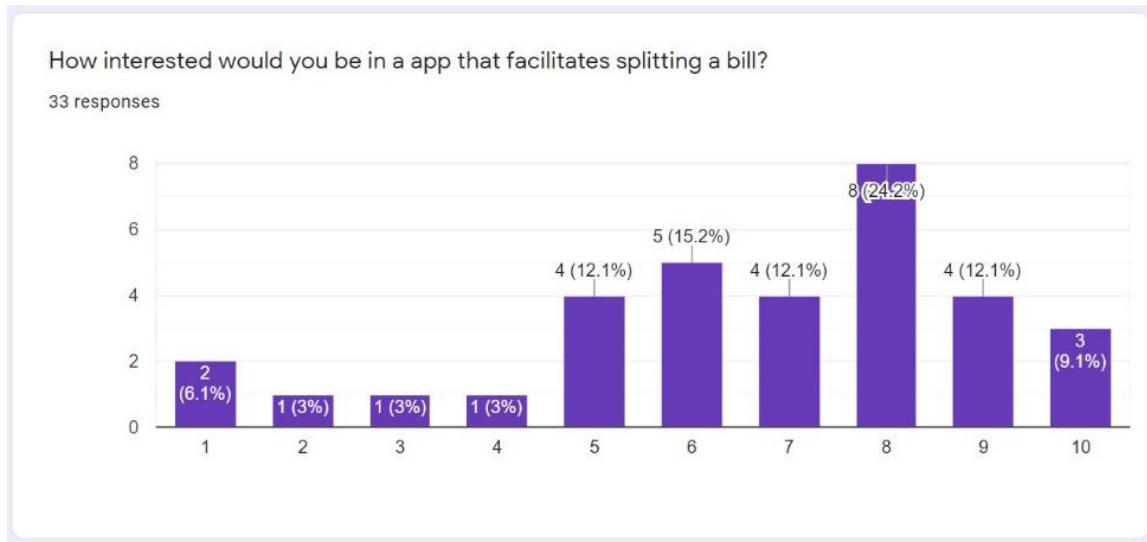


Figure 2: Results to the question asking how interested participants would be in this app.

The ability to choose to pay for specific items gained the most interest compared to the ability to use Google/Apple pay, allowing us to put most of our focus on this feature.

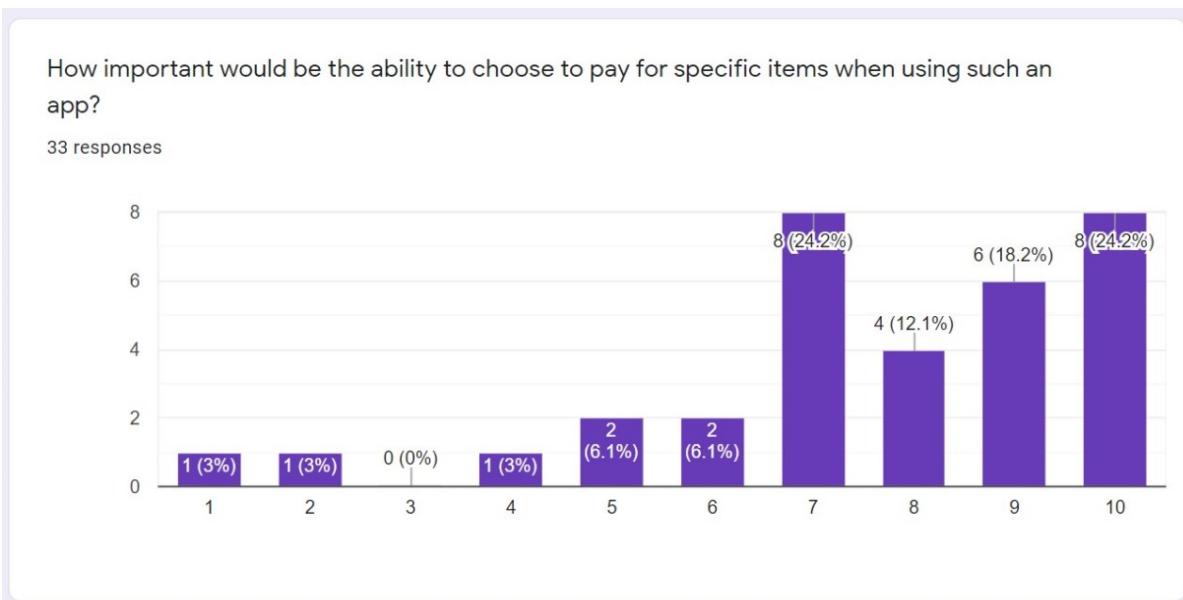


Figure 3: Results to the question asking how interested users would be in choosing their own items.

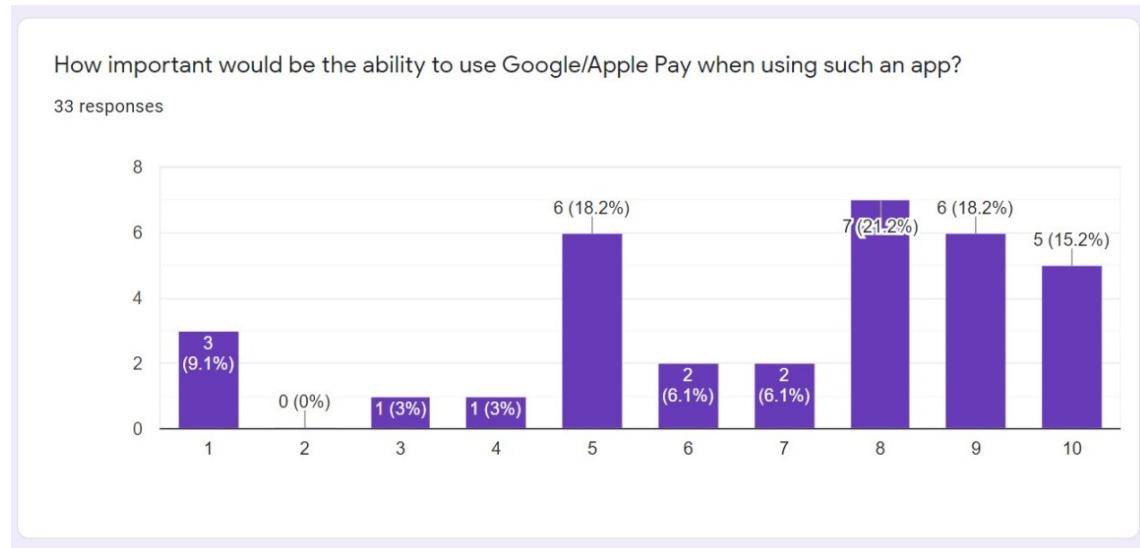


Figure 4: Results to the questions asking how interested users would be in Apple/Google Pay.

Here you can see all the most common answers to the question in this survey:

Question	Most Common Answer	Percentage of Answer
Which type of bills are you most likely to be splitting?	24-40	45.5
How often do you go out each month? (Before Covid-19)	1-4 times (around once a week)	60.6
How often would you split a bill per month? (Before Covid-19)	1-4 times (around once a week)	57.6
Which type of bills are you most likely to be splitting?	Events with friends	78.1
How do you currently split bills? Select all that apply.	Bank Transfer	66.7
How interested would you be in a app that facilitates splitting a bill?	8	24.2
How important would be the ability to choose to pay for specific items when using such an app?	7 and 10	24.2 each
How important would be the ability to use Google/Apple Pay when using such an app?	8	21.2
Which layout do you prefer?	Option 1	71
Please select the colour scheme you like the most:	Light Green	45.5

Figure 5: Most common answers to secondary market survey

From the data on figure 5, we can deduce that our target audience are individuals 24 to 40 years old who go out 1 to 4 times a month and split bills nearly every time they go out. Most bills are split during events with friends via bank transfer. This data indicates that individuals would be interested in using a bill splitting application and would see the ability to spit bills equally as most important.

The individual breakdowns for these questions can be found in the appendices.

Design

Normalization

Normalization is the procedure of efficiently organizing data in a database with two purposes: remove redundant data (as in storing same data in different tables) and logical data dependencies (stored data in a table are related) [10].

This technique could considerably improve the performance of our DBMS due to more speed, accuracy, and efficiency. Additionally, the use of optimal structure composed of elements that cannot be broken into smaller parts (also called as atomic elements) and data being arranged into tables and columns, with each table only containing related data [11].

Guidelines to guarantee database normalization (1NF to 5NF) [10]:

- Guidelines are cumulative: to be considered 2NF, must fulfil all criteria of a 1NF Database
- First Normal Form (1NF)
 - Eliminate duplicative columns from the same table.
 - Create separate tables for each group of related data and identify each row with a unique column or set of columns [the primary key (*uniquely identifies each record in the table*)].
- Second Normal Form (2NF)
 - Meet all the requirements of the first normal form.
 - Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
 - Create relationships between these new tables and their predecessors using foreign keys (*a field in a relational table that matches the primary key column of another table*).
- Third Normal Form (3NF)
 - Meet all the requirements of the second normal form.
 - Remove columns that are not dependent upon the primary key.
- Boyce-Codd Normal Form (BCNF or 3.5NF)
 - Meet all the requirements of the third normal form.
 - Every determinant must be a candidate key (combination of attributes that uniquely identify a database record without referring to any other data).
- Fourth Normal Form (4NF)
 - Meet all the requirements of the third normal form.
 - A relation is in 4NF if it has no multi-valued dependencies.
- A table contains only one primary key, but it can contain several candidate keys.

UML Diagram

This is the final version of our UML diagram which details the design of our system:

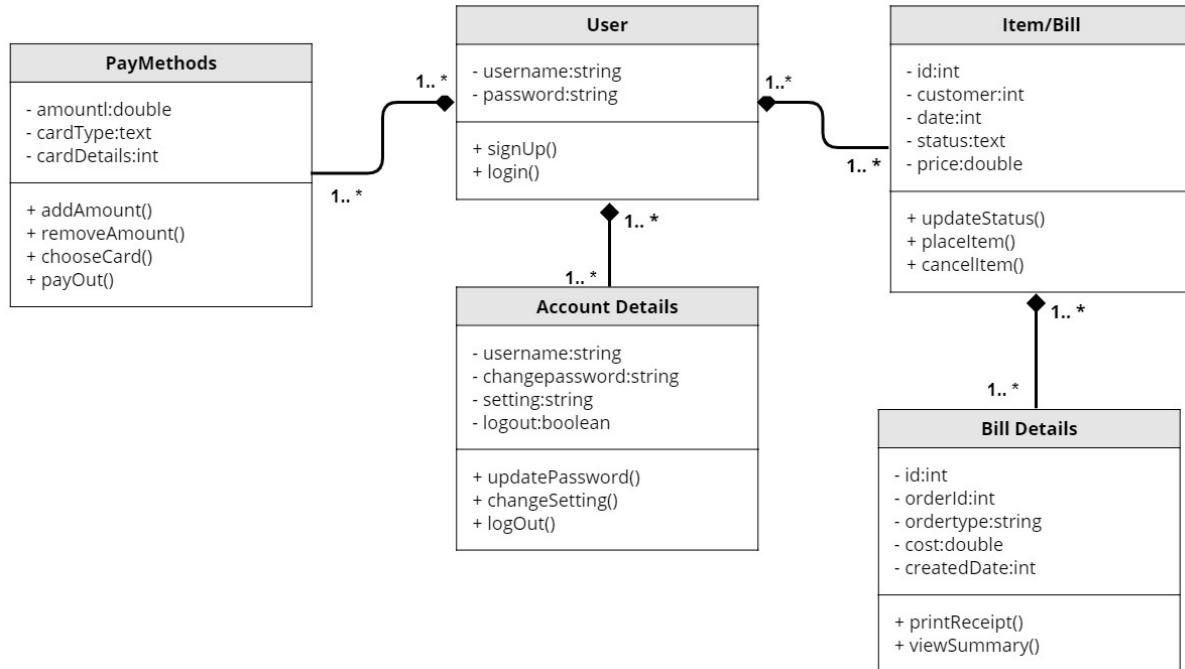


Figure 6: The final UML diagram

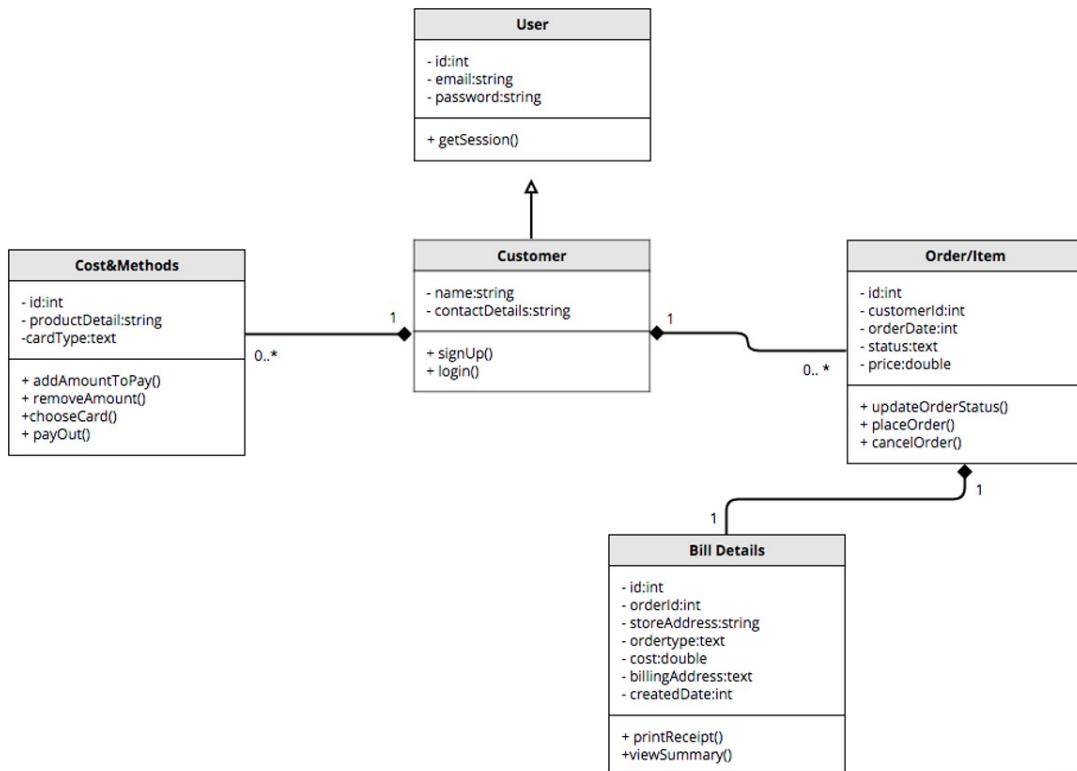


Figure 7: The initial UML diagram

Sequence Diagram

These are the final versions of the sequence diagrams for the bill creator and member, showing the iterations between the application interface, payment gateway and database:

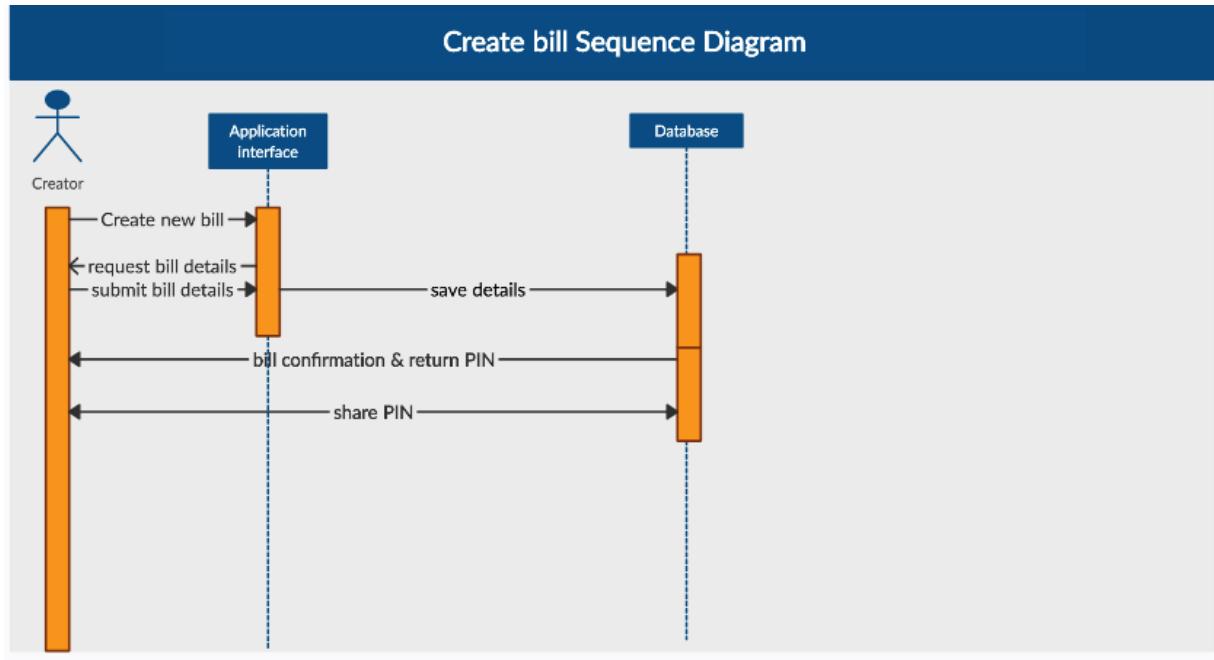


Figure 8: Final sequence diagram for the bill creator

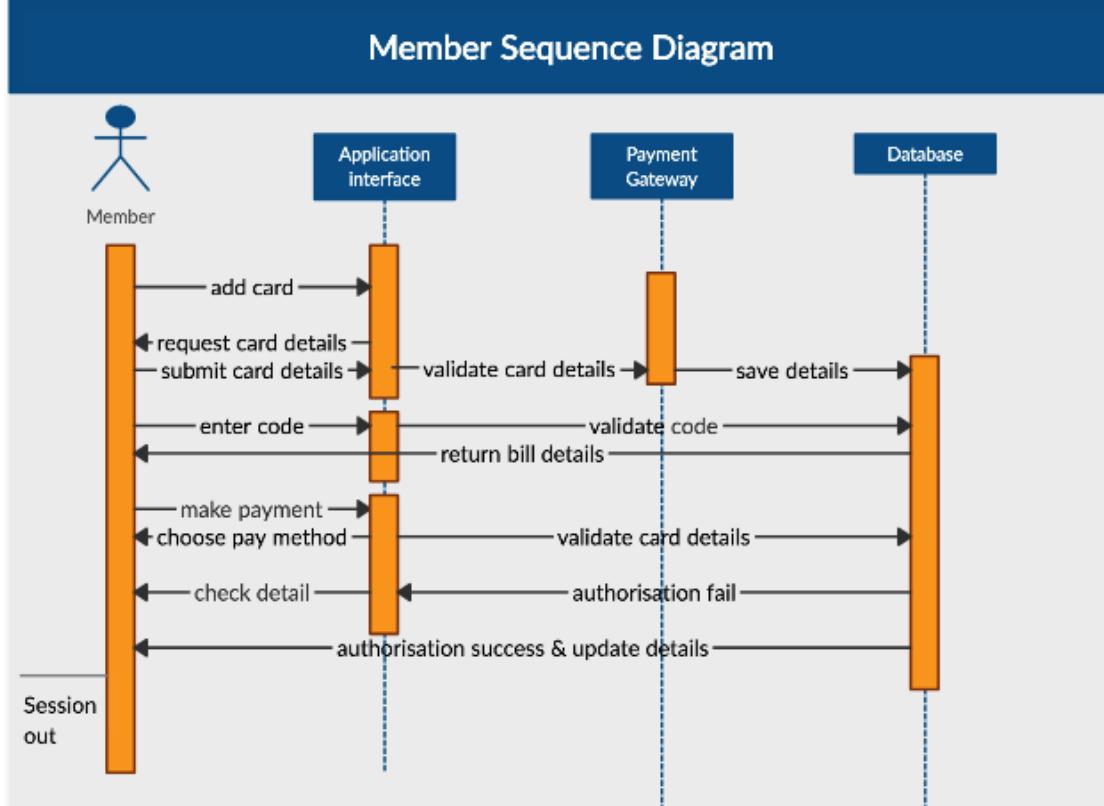


Figure 9: Final sequence diagram for the bill member

This is the original version of the sequence diagram above:

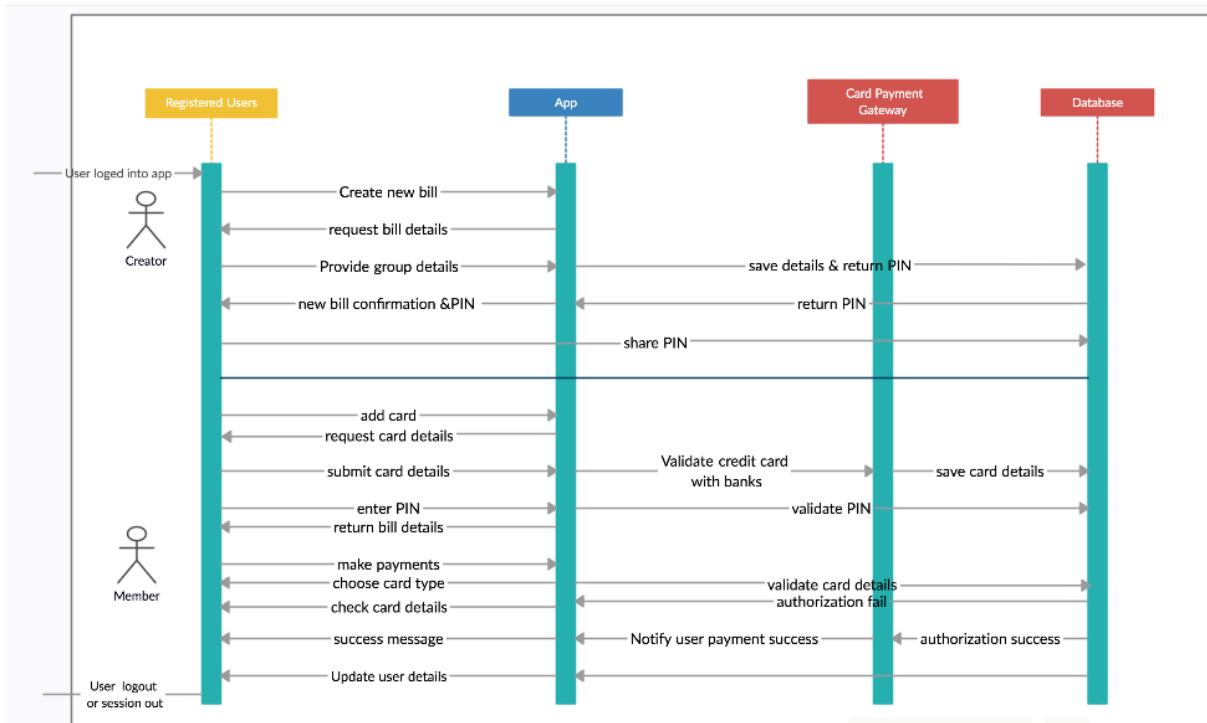


Figure 10: Original sequence diagram

Activity Diagrams

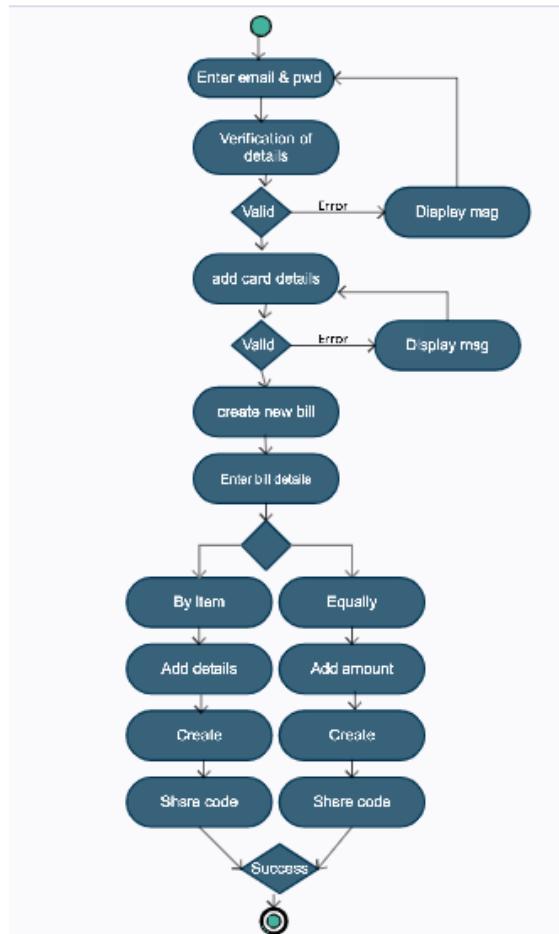


Figure 11: activity diagram for bill creator

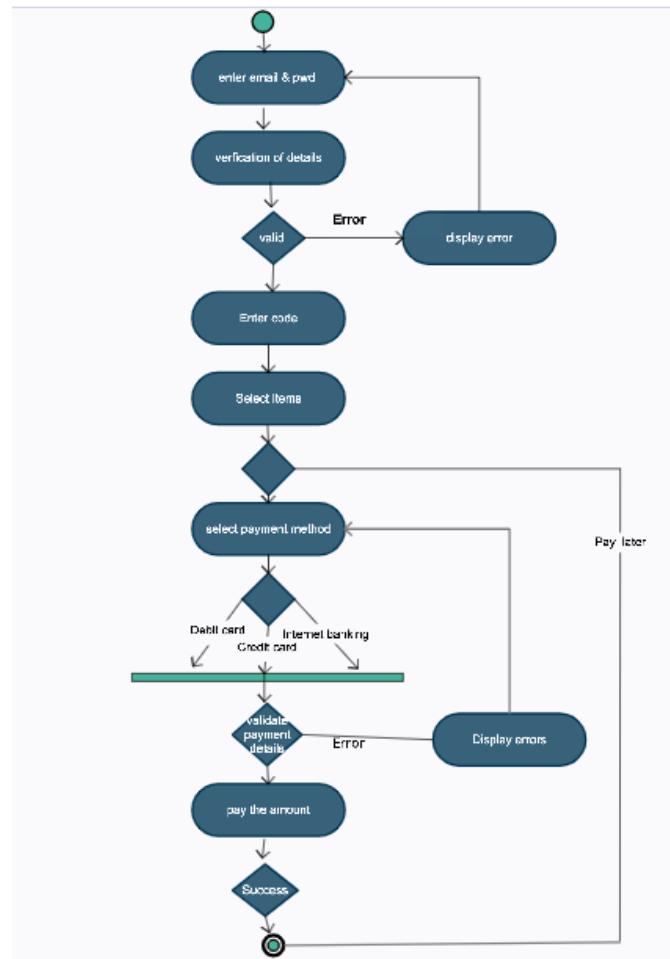


Figure 12: activity diagram for bill member

Use-Case Diagrams

This is our final use-case diagram which shows both the bill creators and bill members interaction with the application when creating/joining a new bill and the relationship between them:

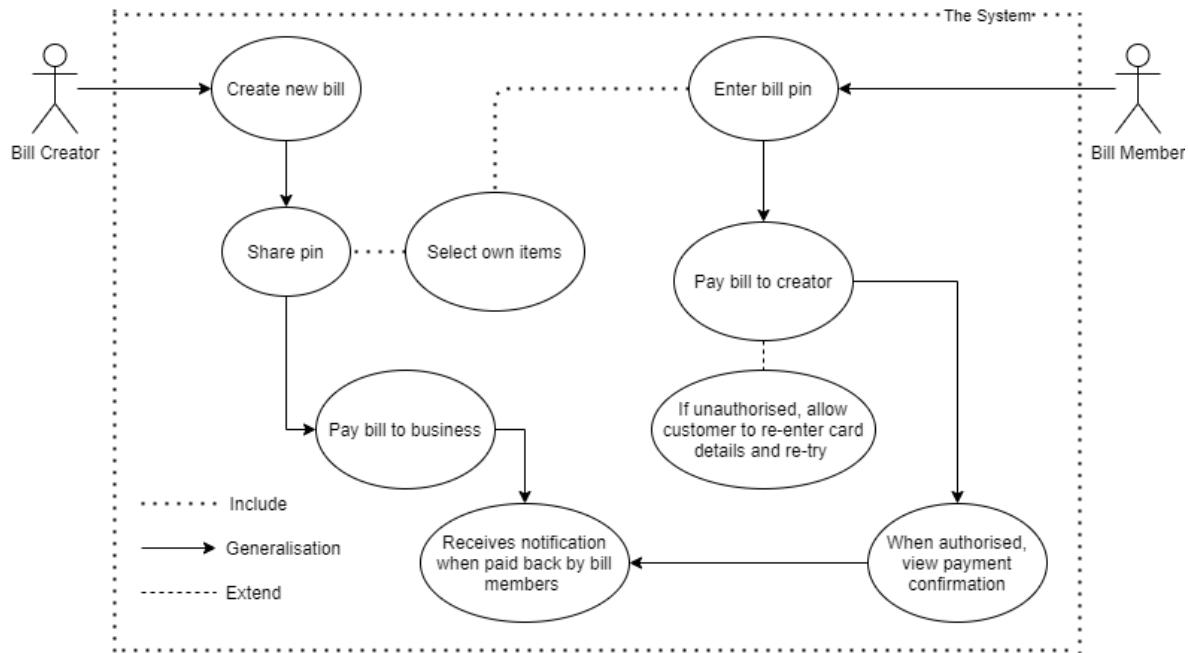


Figure 13: Final use-case diagram

Here you can see the previous version of this diagram:

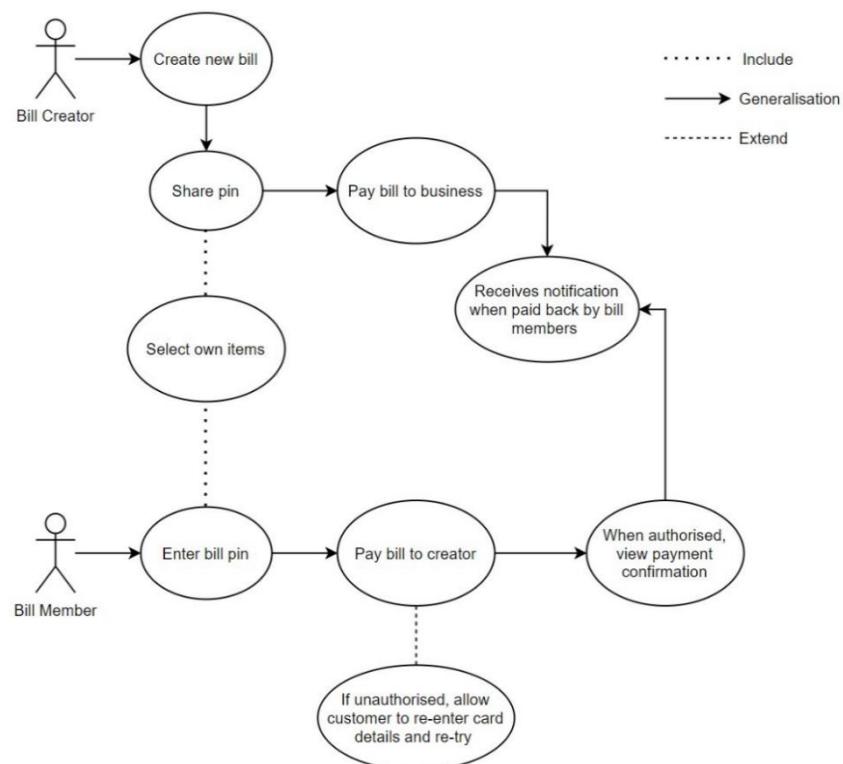


Figure 14: Penultimate use-case diagram

User Journey Mapping Diagram

This diagram shows the journey the user will take from beginning to end and outlines any problems they may run into.

	AWARENESS	SERVICE			LOYALTY
Timeline	Step 1	Step 2	Step 3	Step 4	Step 5
User Thinking	User is having to split a bill with a group of people and does not want to have to share sensitive bank account information with everyone.	Each member of the group needs to be paying a different amount as they have each consumed different items.	User cannot trust that all group members will remember to pay them back soon.	User is worried that some members might accidentally not select one of their items and that they will have to pay for it instead.	User has now been paid back by all the bill members and is happy with their experience using the application.
Touchpoint	User opens the application and goes to create a 'New Bill' to make splitting the bill easier on the whole group.	The user selects 'By Item' for the split type, scans the bill and checks that all items have been scanned correctly. Group members then select the items they should be paying for.	User selects a pay-by date; when users join the bill, they agree that they will pay their share by this date and that if not the application can automatically take payment on their behalf.	Payments from each member will be "pending" until all items have been selected; if all items haven't been selected by the "pay-by-date" users will receive a notification asking them to double check the items they have selected, which will make sure all the items are selected.	User submits a review on the application store recommending the application to other users. The other group members, who enjoyed using the app, also start using it regularly.
Channel	Mobile application.	Mobile application.	Mobile application.	Mobile application.	Application store.
Experience	User experience is positive; they have an easy way to split their bill without having to share sensitive bank account information.	User experience is positive; they do not have to calculate individual bills, and instead the application does it for them.	User experience is positive; they do not have to worry about when their group members will pay them back.	User experience is positive; they know that the application will remind bill members to make sure all their items have been selected if there are unselected items left.	User and the other bill members have had an overall positive experience.

Figure 15: User journey mapping diagram

Technical & Functional Specification

Purpose

The purpose of this application is to facilitate an easier way for groups to split common bills. We expect to create a convenient and easy-to-use application to split bills within groups, allowing users to pay in the moment or later.

Scope

This application will allow groups to split bills in different ways using their Android device. We have developed this application assuming that users want to pay for their share of a bill and so all features put in place is to help them do so. This includes allowing users to split bills equally, which would be necessary for a sharing platter situation and splitting bills by item. While this application will automatically take payment if the user has not already paid off a bill by a certain date, it is not designed to make users pay but rather to simplify the payments process between them. The scope does not include protecting the bill creator by making sure all the items are paid for; by initially paying the bill the creator takes responsibility for the whole bill and the application is only tasked with facilitating repayment.

The group member who paid the bill will create a bill and enter its details, including the number of bill members, a pay-by-date, and the split type. The creator will then be presented with a unique code which will be used by bill members to join the bill. All users must create an account and enter their bank card details first to use the app. The bill creator must enter their bank account details also to accept payments. After bill members join the bill, they have until the pay-by-date to pay back the creator; after that point, payment will be automatically taken.

System Overview

We have decided that we will be making a native application for the Android operating system. Our decision stemmed from the fact that a native application would allow us to program more functionality, such as using the camera, and is generally faster and more responsive than hybrid apps. We decided to make this an Android application since as a group we have more experience in Java than Swift and due to the time constraints in the development of this application we would not have time to learn a new language, so we will be programming in Java SE 15.

According to the data protection act 2018, we must follow "data protection principles" while using personal data. Therefore, the only user with valid login details will be able to access the application. We have also considered using the fingerprint feature to ensure data protection against unauthorised access or processing [12].

Normalised databases such as MYSQL require joining tables to fetch queries which slow down the query as many tables need to be addressed and aggregated before calculating values promptly. Contrarily to this, denormalization databases such as MongoDB speeds up the data retrieval process as it produces calculated values as

it stores data in JSON-like documents and there is no need for structuring documents [13].

MongoDB Atlas, a fully managed cloud database, accelerates building applications with scalability, flexibility, and reliability [14]. It is convenient to create and maintain reports without understanding the application's code once the values are generated and added to tables [15]. MongoDB provides security features such as authentication, access control and encryption. which helps to protect all the personal information.

For our database, our research has led us to the decision to use MongoDB since our application will mainly use transactional queries, for example, as users select items in the bill to pay for, the database will have to make it unavailable for other bill members. This leads us to use a row-based data storage type, of which MongoDB is one we have the most experience with. Due to the time constraints in the development of this application this is a necessary choice, as learning a new system may take too much time.

Payment API

Payment APIs are designed for managing payments by integrating multiple payment sources and providing customers a way to track payments [16]. For every payment, a request is generated. The Payment Request API is "*an open and cross-browser standard that replaces traditional checkout flows by allowing merchants to request and accept any payment in a single API call*". [17]

As a group we aim to facilitate at least two API's for a greater user commodity. Noodlio Pay - Smooth Payments with Stripe, is a free API that replaces the server-side and offers a service that can immediately start marking requests, therefore being promptly implemented. Google Pay is a natural choice for belongs to Google, creator of the Android OS therefore is a fast and simple way to conveniently "*access hundreds of millions of cards saved to Google Accounts worldwide*" with full system integration [18].

Camera API

We will implement the CameraX Jetpack library over the Camera2 API as it is compatible with Android devices running API level 21 and hardware devices with diverse camera configuration or device-implementation details. Camera2 API is powerful but due to the variety of camera capabilities offered by different devices such as HDR or night mode, it can be tricky to get most of it. On the other hand, CameraX helps to enable more camera features in the application. It supports zoom control, device rotation information and ultimately helps users to capture high-quality images and save it to disk [19].

Functional requirements

General

Sign Up	The application should allow the user to input personal data and validate When the register is validated the system will store the data in the system database
Sign in	The system should allow user input username and password to access the system.
Log out	Any user can log out at any point, only by clicking log out button.

Bill actions

Create New Bill	The system should allow the user to input the current event and validate it	
	Split type equally	The system should calculate the amount for payment per user.
	Split type by item	The system should allow the user to choose between scanning the receipt using their camera or adding details manually.
		The system should allow the user to select items they want to pay for.
The system should automatically generate a random unique code to be share with the rest of the group.		
Join Bill	The system should allow the user to input the code.	
	The system should not allow the user to continue without accepting terms and conditions.	
	The system should allow the user to select items they want to pay.	
	The system should allow the user to choose between pay now and pay later	
	Pay now	The system should show the payment methods available on database.
	Pay later	The system should allow the user to review the bill on the past bills page
	The system automatically will validate the others users' payments receipt	

Navigation Bar

Past Bills	The system should display the bills history in priority order		
	The system should display the receipts in date order.	Pending - the system should display the receipts pending for payment	
		Repayments Pending - the system should display the receipts pending for re-payment	
Account	The system should display the user account details		
	The system should allow the user to change password		
	The system should allow the user to add and edit bank card details		
Settings	The system should allow the user to change the colour scheme to High contrast mode		
	The system should allow the user to select "Dark Mode".		
	The system should allow the user to select "Enable fingerprint"		
	The system should allow the user to select "Export data".		

Non-Functional requirements

Security: The application needs to be secure as it will contain users' bank details. Users must enter their password or use biometrics before each time they use the app. This should also be true for when users are making payments.

Capacity: The system software should withstand more than 300 simultaneous users without experiencing the server crashing or significantly slowing down. Such a design can be architected with high-end web servers and databases.

Usability: The application must have a simple, attractive, and intuitive graphical user interface. It is important that users are able to understand the flow of the application without the need for instructions or manuals. This should be accomplished by using simple language, including informative icons as a second form of communication, and having an intuitive layout.

Scalability: The maximum number for members per bill should be high. This will allow users to use the application in most settings and will increase the app's benefits to users.

Network Coverage: In situations where users do not have access to the internet, they should still be able to use some aspects of the application while offline. This includes viewing their account information and the details of their 10 most recent past bills.

Responsiveness: The application needs to be responsive to any outside interruption, meaning it should be on the same state when the user returns to it, assuming that the application has not been closed in the meantime.

Accessibility: The application must be accessible for those with disabilities also. It should include features to make the application accessible for those with a language barrier, colour blindness or dexterity challenges.

Development Methodology

We used Agile methodology when working. While using a Waterfall methodology would have been faster and easier to manage, it would have made it very difficult to make changes in previous phases [20]. Doing simultaneous, incremental work allowed us to do user testing at all points of our project, meaning we were able to adapt the design of our application according to what worked for our stakeholders the best.

We felt that agile was the best choice as we wanted to work in an efficient way that would let us adapt our individual ideas into a well-rounded application that would have high user satisfaction [21]. Agile allowed us to do so. Its collaborative nature meant we were able to work more efficiently than Waterfall would have allowed, as we did not have to wait for certain tasks to be completed before starting the next. Agile methodology enforces adaptability, so we were able to continuously adapt our work to better fit our vision and meet user's needs. For example, by using Agile we were able to conduct testing on version 1 of our high-fidelity prototype and then make the relevant changes to develop version 2, which would not have been possible using the Waterfall methodology.

The type of Agile methodology we used was a mix between Scrum and Kanban [22]; each distinctive stage of development was broken down into sprints, all of which contained multiple milestones and sub-tasks which were ordered based on completion status and priority. We planned the sprints in advance, along with the milestones and sub-tasks for each one and held at least one meeting a week to discuss any problems and work collaboratively. Using this methodology, we were able to plan our next steps based on which milestones were already in progress, which had the highest priority, and which needed amending.

Impactful Supervisor Meetings

Meeting 2

Supervisor: Sean McGrath

Attendees: David Cardoso, Mary Sousa, Yasmin Paksoy

In this meeting, we presented the initial versions of the UML and use-case diagrams we had developed. We also presented the questions and responses to our initial market survey.

Suggestions made by supervisor:

- The use case diagram is too complicated; should only have 4 steps, calculate the total, average it, assign responsibility for overall payment, and be paid back.
- The class diagrams should have more dependencies; bill and methods have dependencies with bill details that are not represented. The cardinality should also be one to one or more.
- The demographics on the survey should be split into logical categories; the age groups should have an attached meaning. This can be done using statistics on which age groups eat out the most often etc.

- Should also be asking how often individuals go out in the survey so it is possible to analyse how often people need to split bills regard to how often they go out; if they never go out, they will never be in a social situation where they have to split bills for example.

Changes made due to comments:

- We remade the use-case diagram, keeping it much simpler. We merged the use-case diagrams for the bill creator and member to make the relationship between the two more comprehensive.
- We edited the class diagrams to represent all the dependencies between classes and edited the cardinality also.
- We researched how often different age-groups went out to restaurants and developed a new survey where the demographics represented these age-groups, adding meaning to our results.
- In the new survey we created, we added a question asking participants how often they went out each month along with the question asking them how often they split bills per month.

Meeting 5

Supervisor: Sean McGrath

Attendees: Bishal Rai, David Cardoso, Mary Sousa, Yasmin Paksoy

In this meeting we presented the initial version of our high-fidelity prototype.

Suggestions made by supervisor:

- The number of members option would work better as a drop-down list. Also, it is not clear is this is the number of members including or excluding the bill creator.
- The "Pay by item" button should sound consistent to the "Split Equally" button.
- High contrast can be a toggle option, it does not have to be the default colour scheme, so we can keep aesthetics while making sure the application is accessible.

Changes made due to comments:

- The number of members option would work better as a drop-down list. Also, it is not clear is this is the number of members including or excluding the bill creator.
- The "Pay by item" button should sound consistent to the "Split Equally" button.
- High contrast can be a toggle option, it does not have to be the default colour scheme, so we can keep aesthetics while making sure the application is accessible.

Meeting 6

Supervisor: Sean McGrath

Attendees: Bishal Rai, David Cardoso, Mary Sousa, Yasmin Paksoy

In this meeting, we presented the first version of our complete high-fidelity prototype and discussed our ideas for testing.

Suggestions made by supervisor:

- Alternative option should be made for if users do not want to use biometrics on the app.
- Filter for past bills would be useful.
- Should test efficiency of the prototype, suggested conducting timed tests.
- Remote usability testing can be done due to the current social distancing measures in place.
- Should also test effectiveness, can be achieved by giving users a set of tasks to do.
- Should check reliability of any testing; learning effect and pressure could affect results.

Changes made due to comments:

- We added a login option for users who deny the use of biometrics to the prototype.
- We added a filter option to the past bills page on the prototype.
- After the prototype was complete, we conducted timed tests to assess efficiency.
- We also did remote usability testing to assess effectiveness by asking participants to complete a set of tasks.
- While designing both tests we made an effort to ensure reliability and discuss this more in the prototype section.

Prototyping

Low-fidelity prototyping was a quick and easy way to translate our design ideas and concepts into a perceptible and demonstratable image. Since this type of prototyping supports rapid experimentation, we were able to create multiple designs, allowing all members to adequately present their ideas and giving us the opportunity to debate and analyse their benefits and drawbacks as a team. Only a few features of the low-fidelity prototypes ended up being included in our high-fidelity adaptation but exploring different concepts and refining designs speedily allowed us to be confident in our design choices. While it is not possible to deliver complex animations or structure using such prototypes, it allowed us to further develop our design ideas in a systematic way.

During the development of low-fidelity prototypes, we knew that the navigation of the application needed to be intuitive and predictable, making it simple for users to discover how to move in the different sections with comfort. We experimented with different menu structures, such as the hamburger menu, one of the most common structures to save space, as seen as in figure 16. This menu style hides the navigation on one side of the screen and is displayed only after a user action, and while it saves space it does not communicate the current position to the user.

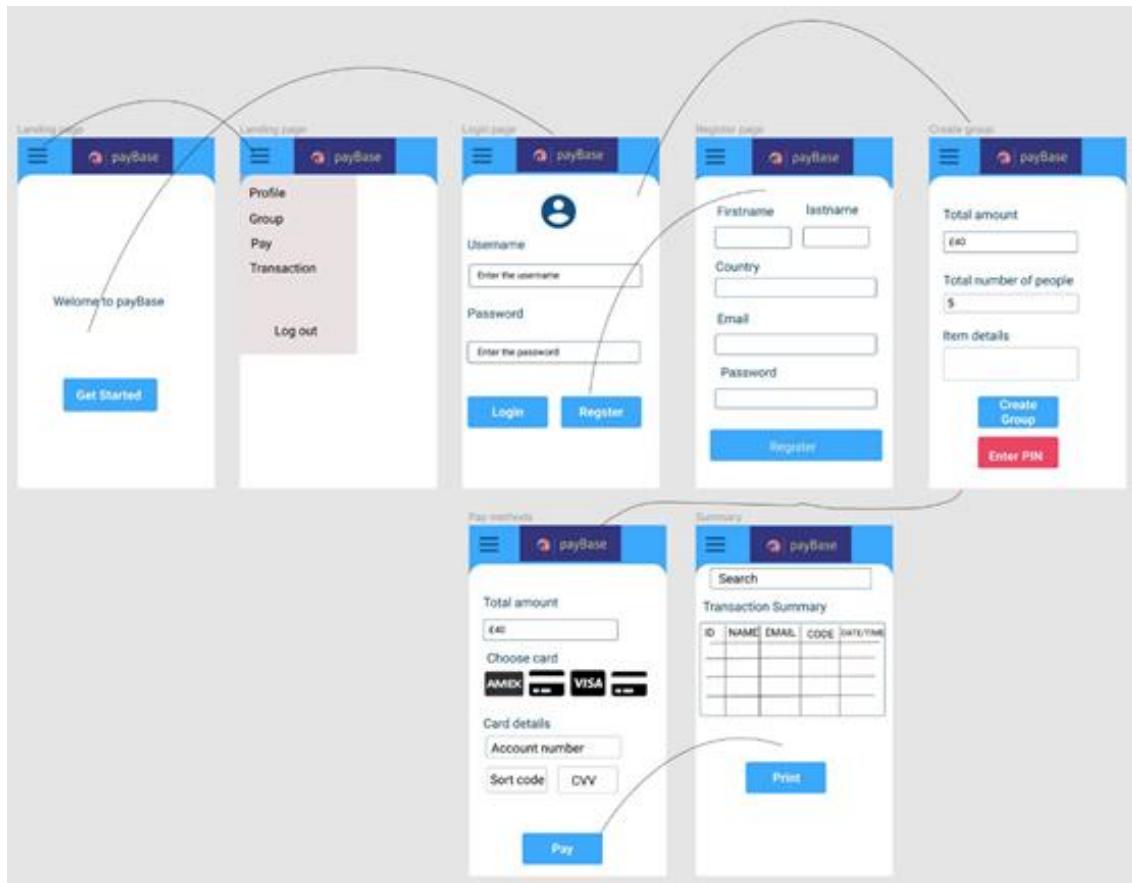


Figure 16: Low-fidelity prototype with hamburger menu

We experimented with using a tab bar because it has a limited number of clear priority options to improve user's navigation and give direct access from any part of the app, quickly and easily communicating the current position of the user, as shown as on figure 17.

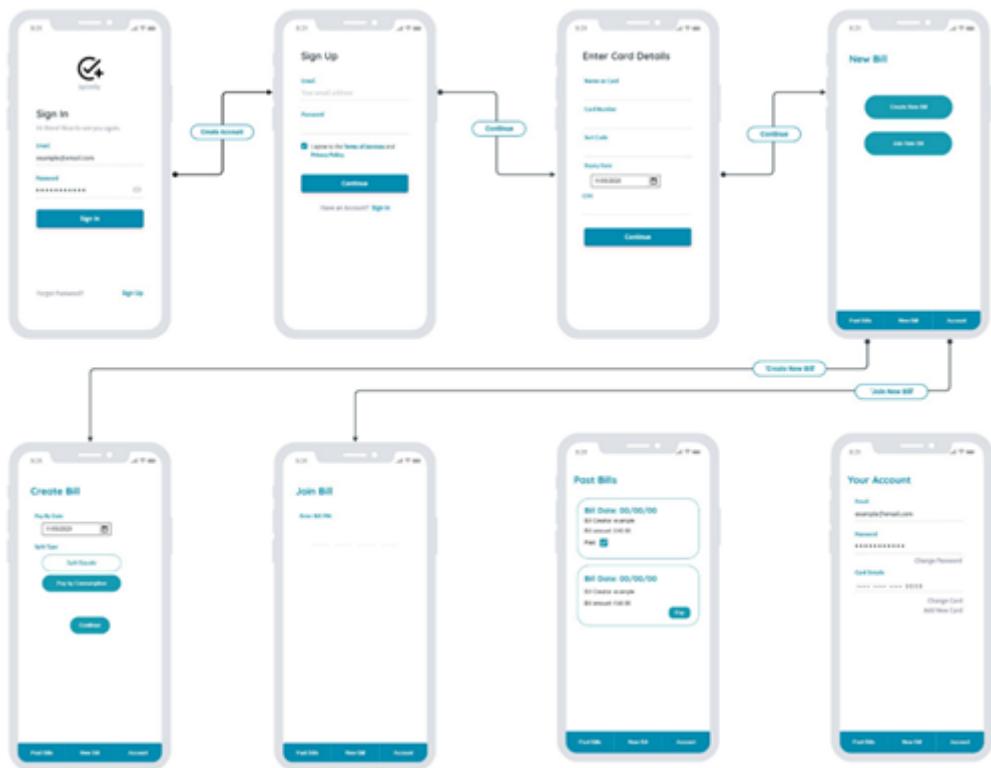


Figure 17: Low-fidelity prototype with tab bar menu

Experimenting with different menu structures also gave us an opportunity to discuss system flow, which we later developed in the high-fidelity prototype.

We also trailed different design ideas to allow us to consider which style looked aesthetically pleasing, as you can see on figures 18, 19 and 20.



Figure 18: Low-fidelity prototype for design 1

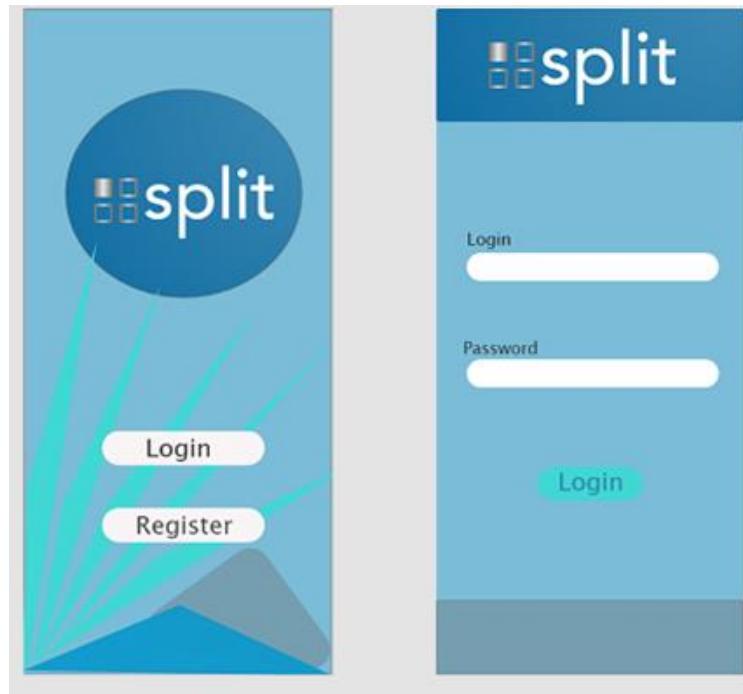


Figure 19: Low-fidelity prototype for design 2

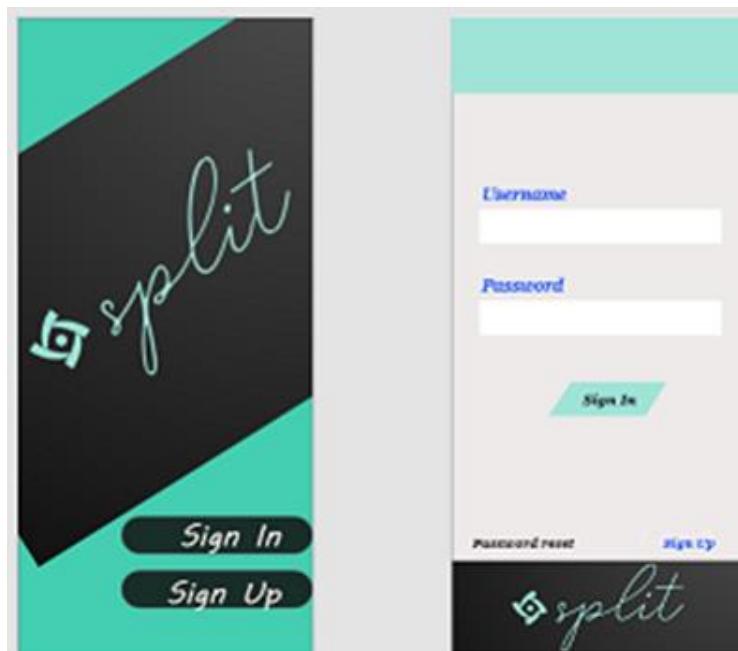


Figure 20: Low-fidelity prototype for design 3

Based on these wireframes, we were able to make a functional high-fidelity prototype using Adobe XD.

For high fidelity, we choose what we believed to be the best aspects of the different low fidelity prototypes we had developed. Most decisions were based on the current convention regarding the user interface of most applications in order to reduce time users would have to spend learning and eliminate any confusion they could encounter [23]. For example, we decided to place the navigation menu on the bottom of the screen, as shown in figure 21, as that is the current convention. This also has the benefits of making it easier for users to find common features;

for example, the application Redbooth found that session time increased by 70% and daily active users increased by 65% when they switched from a hamburger menu to a navigation bar [24].

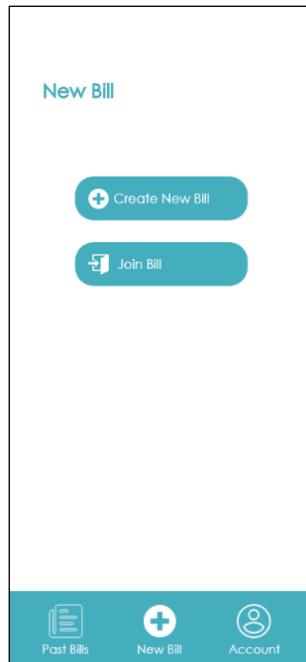


Figure 21: Splitsy home page

Our colour choice was based on both aesthetics and colour psychology. We found that the colours blue and green best represented the feelings we wanted to bring forth in users; blue invokes trust, dependability, reliability, and calmness while green invokes balance, harmony and relieves stress [25]. Due to these factors, and because we found this colour to be aesthetically appealing, we choose turquoise, a mix of blue and green. We decided to have a monochromatic colour scheme to ensure the application looked visually cohesive and to help our users build an association between our application and the colour turquoise.

When deciding the layout, we wanted to know what felt most attractive to potential users, so we included a question with two layout options in our secondary market survey. An overwhelming majority chose option 1, as shown below, so we applied that layout to the main page of our app.

Which layout do you prefer?



Figure 22: Question asking users which layout option they prefer

Which layout do you prefer?

31 responses

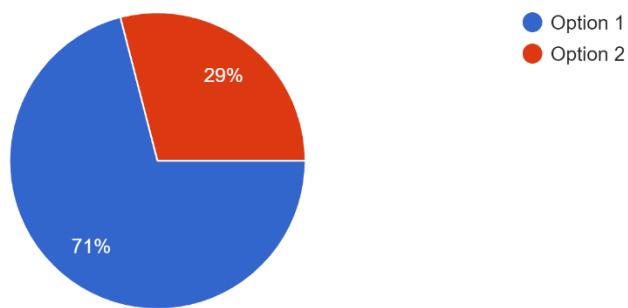


Figure 23: Results for the question asking which layout participants prefer

Once we finished all the pages for the first version of this prototype, we added links between the pages, so the buttons then took the user to the correct page, as shown as in figure 24.

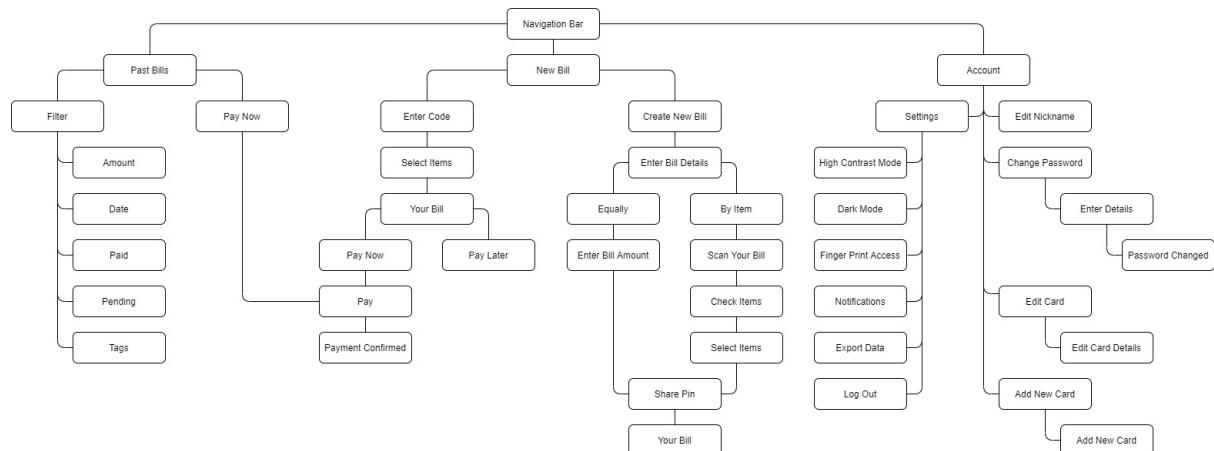


Figure 24: Hierarchy of pages on version 1 of the complete prototype

Timed Tests

After creating the first complete version of the app, we conducted timed tests. Had it been possible, we would have conducted this test on a diverse group of people of all ages and backgrounds. However, due to COVID-19 pandemic, we had a very limited number of potential participants.

We asked participants to carry out a set of 10 basic tasks, as shown in figure 25, which we believed to be part of our minimum viable product. We timed participants during each task to verify that the application had a logical navigational structure and that features were not located in elusive locations. Results of these tests are shown in figure 26. At the end of the timed test, we asked users for any comments they had regarding their experience, which can be found on figure 27.

1. Sign up and enter bank details
2. Login and change your password
3. Create bill which is split equally and continue to the share code page
4. Check past bills and pay the pending bill with cash
5. Add new bank card to your account
6. Join bill, select the first item and complete payment with card
7. Logout
8. Login again and enable fingerprint
9. Enable high contrast mode
10. Create bill, which is split by item, scan a receipt, and continue to the share code page

Figure 25: List of tasks participants were asked to complete in the timed tests

Participant ID	Age	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
P1	18	78	38	73	20	30	45	40	15	13	52
P2	19	55	50	55	39	59	47	82	45	18	72
P3	18	15	18	17	14	10	44	6	5	8	24
P4	49	41	54	31	27	22	52	30	22	37	48
P5	55	47	63	52	47	42	49	40	34	39	43
P6	34	44	30	40	27	36	31	8	16	10	24
Average	32.2	46.7	42.2	44.7	29	33.2	44.7	34.3	22.8	20.8	43.8

Time is measured in seconds

Figure 26: Table showing results of timed tests

The result table shows the time taken for every task with overall average. Some of the tasks took the participants much longer than expected such as adding card details whereas some of them took short time. We found that participants who have used similar application before found it easy to complete the tasks.

On average, participants took much more time to complete task 2 out of 10. Hence, we analysed the result in our meeting and planned to improve the related tasks to enhance user experience.

Participant ID	Comments
P1	There should be instructions on how to use it
P2	It was difficult to find logout
P3	Instructions not clear for joining bill; say enter code and join bill
P4	Joining bill was difficult
P5	hard to find log out and adding card details
P6	Color scheme too bright; logout could be on account; enable fingerprint page not perceptable at 1st glance

Figure 27: Table showing participants comments after testing

Each participant followed the same order of tasks. While following the same order does mean there might be a learning effect on our end results, we felt it was unnecessary to randomise the order as we wanted to simulate an organic experience for our participants; in real-world situations, tasks would always be completed in a specific order, for instance, logging in first and then creating a new bill. Furthermore, this test only analysis the first interaction the user has with the app; in real-world situations, after the first time an application is used, users are much more confident as they have now learnt how to use the app, so we did not feel it was necessary to control the learning effect.

While most participants were generally happy with the application, there were subtle issues that were brought to our attention; for example, joining a bill was difficult as the button said, "Enter Code", leaving users confused as they were looking for a button reading "Join Bill". Additionally, users were having a hard time locating the "Log out" option as it was in the Settings page and not the Account page. While watching users, we noticed that they were not reading the confirmation message on the Biometrics page, only clicking "Confirm" to be able to move on and that there were certain pages missing.

We implemented changes to fix all the issues brought to our attention including moving the "Log out" option to the Account page, changing the wording on the "Enter Code" button to "Join Bill", adding a tick box to the Biometrics page to draw more user attention and making any additional pages, adding the links to go with them, as shown as in figure 28.

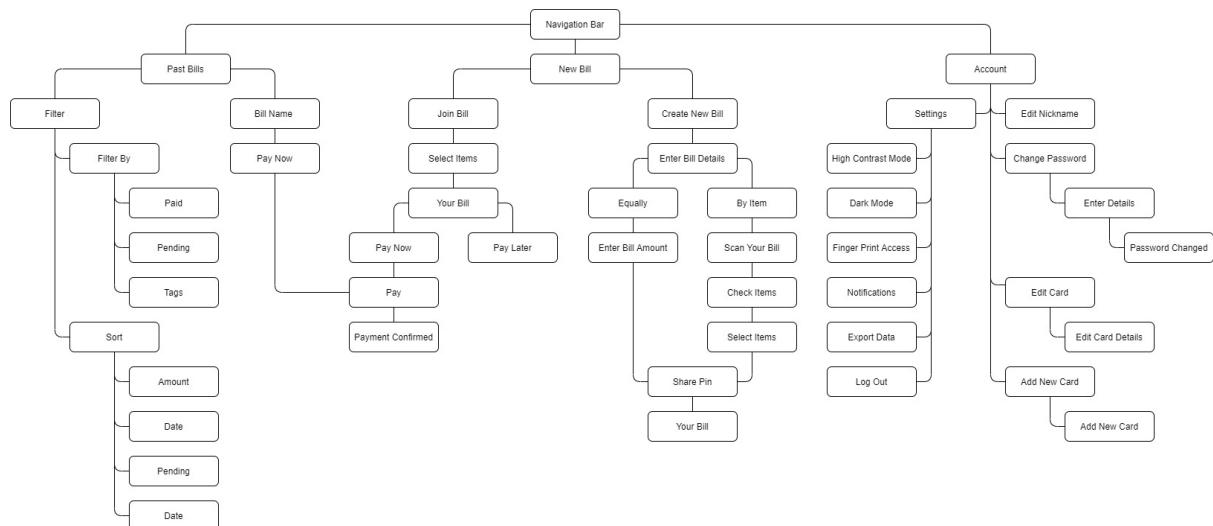


Figure 28: Hierarchy of pages on version 2 of the complete prototype

Survey Tests

To test version 2 of our prototype we made a survey which we shared with friends and on social media. The survey included a link to the second version of our prototype and had a list of 5 tasks we asked users to perform to test that the application was intuitive and had a logical navigational structure. We then asked a set of 6 questions to measure user experience using a 5-point Likert scale. We decided to use a 5-point scale in this survey since this is a more time-consuming survey, meaning fewer individuals would be enthusiastic in completing it and so a 5-point scale would make the results easier to analyse. We also added a textbox at the end to allow participants to leave any feedback or feature suggestions.

We found that most users had a positive experience. The most common answers for each question, shown on figure 29, was 5, the highest possible option.

Question	Answer	Occurrence Percentage
How intuitive was it to know which buttons to click next?	5	36.8
How easy was the colour scheme to read?	5	63.2
How clear were the instructions on the app?	5	47.4
How relevant were the icons to what they are supposed to be representing?	5	52.6
How helpful were the icons when navigating the app?	5	52.6
How easy was it to reach the high contrast option?	5	36.8

Figure 29: Remote testing survey most common answers

Most feedback left was positive, and any suggestions made by users, as shown on figure 30, were sporadic and inconsistent, so we decided it was not necessary to make any more changes to the prototype.

The app should have the option to take a picture of the group

if i don't have biometric device why the payment don't give the option to be a code

Difficulty to find log out

the thing that was hard to find log out

Over all, the app was well presented, but it was a bit difficult to find certain settings.

I had to go 3 or more steps just to logout after payment , what happens if you pick the wrong item before paying(any ways to double check/verify), what happens if I choose pay later and change my bank account or bank balance is 0, I did not see any option to add new card in payment page and I had to go to other page to add new card details, colour was a little bit lighter

The instruction for joining a bill were unclear as it said join bill, but the app said enter code, was hard to figure out they meant the same thing.

Was easy to use, after the first time I would be much more confident

Minor issues but may be due to my personal preference.

The icon for create new bill, caught me off guard as its inverted I thought it was a hospital icon. I'm used to seeing the regular plus. The join bill is quite similar to log in.

I think the buttons "create new bill" and "join bill" may be better in the centre.

Other than those it looks incredible. Also where is Grapes & Bites & Good time.

The app works properly and the colour choice is perfect

Very easy to navigate around

Good and simple

I think the app settings should be contained outside of the Your Account page, since they are two different menus and in my opinion have not much relevance, so. I found reaching options like the high contrast option in the your account page was confusing. Apart from that the rest of the app was well structured, with a very high attention to detail and a great colour scheme. Overall an excellent app with an excellent design.

Figure 30: Remote testing survey feedback box results

Accessibility

Disability is part of being human. More than 1 billion people are estimated to live with some form of disability which corresponds to about 15% of the world's population [26]. Since day one, there has been an intent to create the best application experience for everyone. With this in mind, we have naturally endeavoured all efforts to provide a truly accessible application. After all, accessibility enhancements will not fade the experience of non-disabled users in any way. They too, in fact, will have a much better experience.

Fonts

One of our main goals is to deliver greater legibility, readability and therefore accessibility. For this, we researched fonts and text accessibility [27]. On its latest report in 2018, the W3C recommends that "*text can be resized without assistive technology up to 200 percent without loss of content or functionality*" [28]. So, our application will have relative font sizes to resize appropriately across multiple devices; sentence length, paragraph length and complexity of language also have been assessed for optimized text accessibility.

To be truly accessible, typeface also must be designed in combination with other best practices but without a good choice of typeface things like colour contrast will have limited impact. [29]

Colour Contrast

Through research, we found that 8.6% of the population [30] live with visual impairments and will need extra support to access mobile applications like this project. Having a high contrast colour scheme allows such individuals to be able to see material clearly and to read without putting extra strain on their eyes. This is exceptionally critical for certain individuals with visual impairments as they may be completely unable to peruse information if it is not shown with a high contrast colour scheme. We also found that white background overall context should be reduced to at most 70% of the space as to not produce eye fatigue. After this research, we decided that the system should include a dark mode that changes the background colour and a high contrast option that will change the colour scheme of the fonts and icons.

We decided to include an option for high contrast rather than making the default colour scheme high contrast as we wanted the application to have an aesthetically pleasing user interface and found that most high contrast colour schemes did not allow for that. We chose 4 high contrast colour schemes using the online tool "Coolors". This tool also allowed us to check how these colour schemes would be viewed from the perspective of those with different types of colour blindness, meaning we could ensure that all potential users would be able to see the material clearly. We included these 4 options on our secondary market survey to allow potential users to decide which one we would implement, as shown on figure 31.

Please select the colour scheme you like the most: *

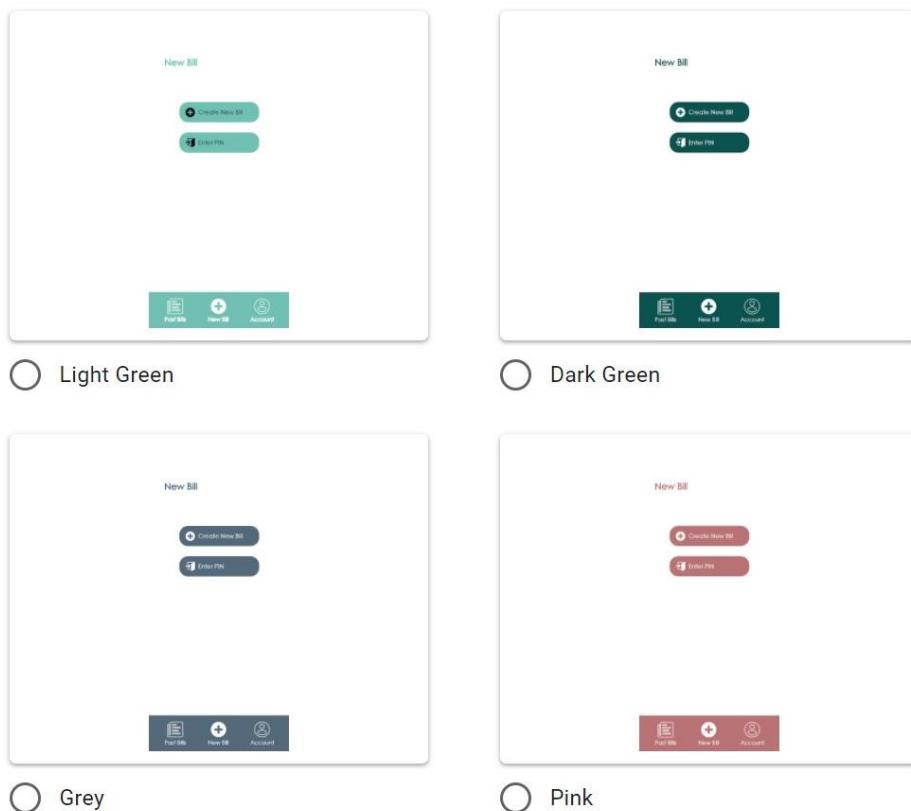


Figure 31: Secondary market survey question 11 regarding colour contrast.

The most voted for high contrast colour scheme was the light green option, as shown on figure 32, so we will be implementing this colour scheme as a high contrast option on the application.

Please select the colour scheme you like the most:

33 responses

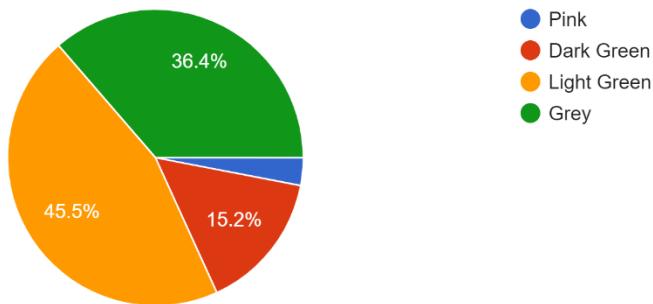


Figure 32: Secondary market survey question 11 results

Language

During research into the different aspects of accessibility, we found that 16.4% of adults in England, or 7.1 million people, can be described as having 'very poor literacy skills' according to the National Literacy Trust [31]. Due to the high number of individuals who face such difficulty regarding literacy, we decided to use several features to make this application more accessible for them. We tried to use short sentences with simple language to better serve these users and included conventionally recognisable iconography [32].

Button Sizes

We did research into the interactions individuals with dexterity issues had with applications to ensure our application was accessible to them. In his research paper, Chen found that "*button size and the presence of disability significantly affect trials with miss, error, and the time to complete tasks on a touch interface*" [33]. This research allowed us to set the minimum button size to 30mm, as he found that the percentage of times disabled individuals missed was the closest to non-disabled individuals at 30mm (roughly 113 pixels). Due to this, we kept the size of the main buttons 44mm wide (roughly 166 pixels), which can be seen on figure 33.

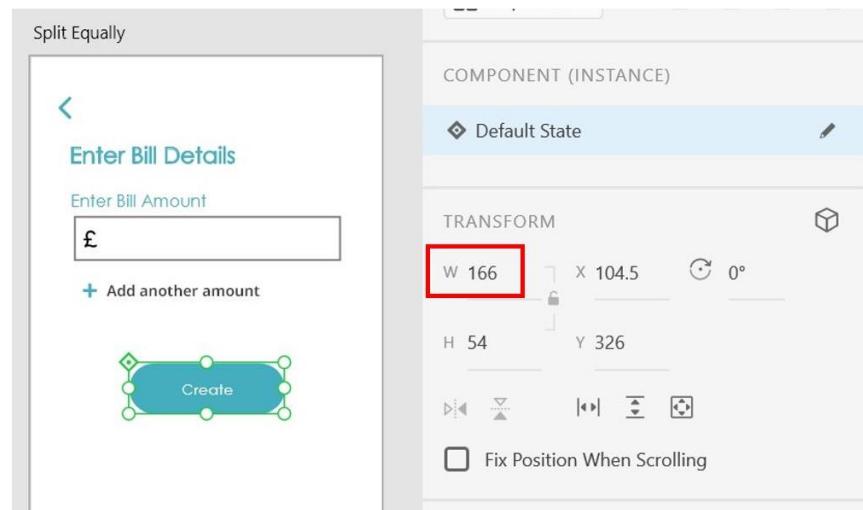


Figure 33: Image showing button size on the 'Enter Bill Details' page.

Project Management

Any project long in time requires the use of well-structured project management tools. As a group we have chosen to use Notion and Gantt chart.

Notion is a cloud-based application that combines a wide range of features into one place. By using Notion, we have been able to effortlessly create, share and track each aspect of this project. Through the use of lists, relational databases, and tables we have created a system of records which is easy to organize, retrieve data from and monitorization. Consequently, Notion has improved the group's efficiency by providing a collaborative hub facilitator while also offering a backup on all the data generated.

The screenshot shows the Notion workspace for a 'Software Project'. The left sidebar lists 'Work Space' sections like 'Roadmap', 'Plan', 'Group Notes', and 'Virtual Contact Notes'. Below that is a 'Shared' section listing team members: David, Mary, Bishal rai, Yasmin, and others. The main area features a large image of a lightbulb on a clipboard next to a laptop keyboard. The title 'Software Project' is at the top, followed by a 'Report' section with a table of contents and sections like 'Admin Details', 'Tasks', 'Milestone Plan', 'Objectives', 'Meeting Details', 'Time Trackers', 'Documents', and 'Report'. At the bottom, there's a 'Admin Details' section and a note about being a guest.

Figure 34: Project home page on Notion

Notion proved to be an essential main tool that, along with the Gantt chart, helped the project to develop as expected, to which every member benefited and contributed.

Objectives

▼ Phase 1: Concept

List scope of achievements and outcomes: Finalise concept, requirements, stakeholders and agree on application flow. Research the market by doing surveys and looking at existing applications.

Specific Objectives:

- Concept research
- Write up final concept
- Write up stakeholders and requirements
- Plan and execute market research

Milestones for objectives:

- M1.1: Research the market, agree on concept and write up.
- M1.2: Research and decide on stakeholders and requirements and write up.
- M1.3: Plan and execute market research.

▼ Phase 2: Elaboration/Planning

List scope of achievements and outcomes: Elaborate on how user is going to interact with the application and plan how to

Specific Objectives:

- Set milestones using GANTT chart
- User journey mapping

Milestones for objectives:

- M2.1: Complete GANTT chart

▼ Phase 3: Design

List scope of achievements and outcomes: Start planning application flow and the different pages the app will have. Complete diagrams to better execute vision.

Specific Objectives:

- Do UML diagrams
- Complete use-case diagrams
- Sequence diagrams
- Class diagrams
- Prototyping
- Create wireframes

Milestones for objectives:

- M3.1: Complete Diagrams

Figure 35: Phases listed on notion with their individual milestones

Figure 35 shows the phases of this project listed on Notion, along with the milestones for each one.

Figure 36: Kanban board of tasks on Notion during research

Milestone Plan		
Plan		
Date	Milestones	Description
Oct 17, 2020	M1.1	Research market, agree on concept, write up
Oct 17, 2020	M1.2	Research and decide on stakeholders and requirements and write up.
Oct 28, 2020	M1.3	Plan and execute market research
Oct 28, 2020	M2.1	Complete Gantt Chart, user journey mapping
Oct 22, 2020	M3.1	Complete diagrams
Nov 11, 2020	M3.2	High-Fidelity Prototypes

Individual Tasks		
<ul style="list-style-type: none"> Write up concept Add technical requirements Execute market research: app flow Do Gantt chart Do use-case for creator Do use-case for past bills Do use-case for bill member Do sequence diagram Create wireframes Do class diagrams Prototypes Prototype accessibility research 		

Figure 37: Milestones listed with their individual tasks on Notion

Figure 36 shows the Kanban board we used to track the individual tasks associated with milestones. This was done using a relational database which allowed us to link each task to its specific milestone, as shown as figure 37.

In conjunction with Notion, we also used a Gantt Chart to manage our project. This type of bar chart represents a project plan in time where tasks are listed, and the workflow is visualized throughout the correspondent bars. The main reason for our choice on this tool resides on its ability to break a vast project into smaller and more manageable tasks. Additionally, the Gantt chart, shown in figure 38, assisted us to visualize the process and progress, serving as a tool to keep the focus on the smalls tasks while maintaining a bird's eye view supporting the entire

project [34]. The project's Gantt chart has been updated over time and now, at the end of Submission One, allows a thorough analysis of resources and time management.



Figure 38: Complete Gantt Chart

It is clear on the Gantt Chart that, overall, the group has performed as expected. The assigned tasks have started when expected, with some tasks even being able to start earlier such as "Build & test prototype". On the other hand, half of the scheduled tasks required more time than projected. Consequently, resources and priority have been reallocated accordingly and the project has been able to recover for this situation. Regardless of delays being a natural part of any project, the use of a Gantt chart revealed to be vital to identify and minimize its impacts.

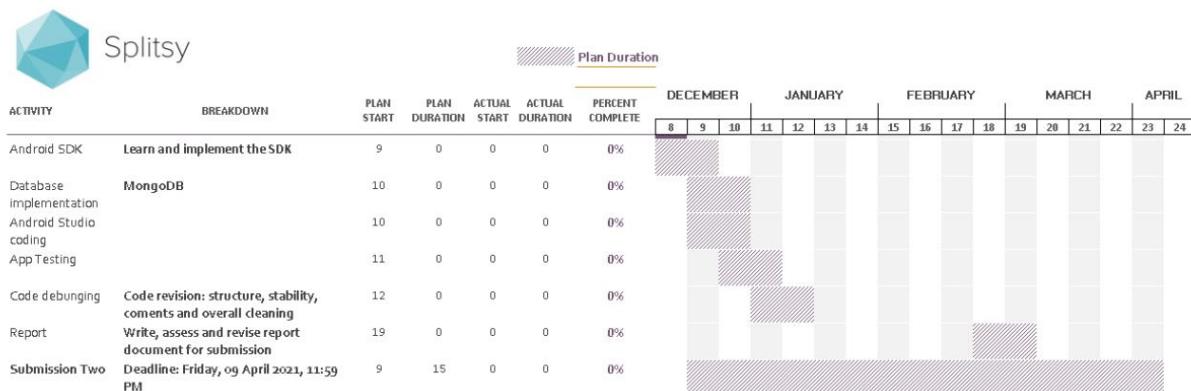


Figure 39: Gantt chart draft for the second half of the project.

Figure 39 shows our prediction for the second part of the project. We will update this Gantt chart once we further define the milestones and tasks.

Lastly, we have used Microsoft Teams as our main communication tool. Teams' main features include chat, threaded conversations, meetings, and video conferencing. The use of Teams revealed to be of great importance since we have not been able to physically meet due to the need for social distancing.

Consequently, a communication tool was key for members collaboration towards the common goal. Additionally, Teams was used as the communication channel between the group and the module leader.

Ethical Audit

This application relies on the ability to store user's personal information, specifically their bank account information, along with other details such as emails and passwords. This is all sensitive information that must be kept secure. This information will be stored in a database. Users will only have access to their own information and to ensure others cannot access this information, even in the instance where they have the users' phone, the application will require the user to confirm their identity every time they open the application using biometrics, such like a mobile banking app. For users who deny the use of biometrics, they will need to enter their password every time they use the app.

The data regarding users create and join bills also needs to be secure; data is very valuable and data pertaining to individuals spending habits is particularly sought after for various marketing purposes. This information will also be stored on a database; no third parties will have access to the data gathered to ensure our user's privacy. The biggest case of fragility is the attack on databases through false identities; this can be avoided with updates and data validations every time the user wants to edit or change any section of their personal data.

All data stored on our databases will be encrypted to stop any attempt of a malicious agent tampering with or stealing any data. When storing data, the system will keep it no longer than what is necessary regarding the Data Protection Act of 2018.

All market research was conducted on adults and no minors were involved. While the main target audience of this application are adults, it can be used by minors, on the condition that they have a children's bank account.

To make our application accessible to as many people as possible, we implemented certain design elements that would minimise any potential problems disabled individuals could encounter; we made sure all buttons were a suitable size, included a high contrast option that would function for most types of colour blindness and included relevant icons to add additional meaning to buttons to minimise any problems language barriers could introduce.

Evaluation Plan

Software testing will involve testing all the essential functionality to meet user requirements such as adding card details to pay the bill. Each team member will perform peer-review for a collaboration strategy. This will allow us to solve related problems and enhance productivity. We will also perform testing through the development phase individually to ensure effectiveness and stability.

These are the tests we believe this application needs to pass in order to be the minimum viable product:

Functional/ Non- functional Testing	Testing name	What is the test about?	Test cases during development	Test cases after development	Measurement during development	Measurement after development
Functional	Unit testing	Validating each unit or component of the software performs as expected, it is carried out by developers.	Test some lines of method or function if it performs as expected such as a sign-up button	Test the whole method or function to verify if the user is signed up	Is the sign-up button's colour, location and shape well managed so that users can sign up?	Does it allow users to sign up and confirm as expected?
Functional	Integration testing	Validating a group of methods performs as expected.	Test pay now button if it allows users to pay their bills.	Test the pay now functionality if the users has paid successfully.	Do integrated systems interact well?	Does the system allow users to pay with ease?
Functional	System testing	Test system's consents with the required specifications.	Test systems ability to meet most of the software requirements such as being able to login, create bills and share the code.	Test entire systems features/functionality to meet all the software requirements.	Does the system fulfil most requirements?	Does the system fulfil all the software requirements?
Functional	Acceptance testing	Test the system's acceptability.	Allow users to test some specification such as	Allow the users to test and verify the entire system to verify the system's compliance with the requirements and delivery criteria for end users request.	Does it meet the requirement specification?	Does it meet all the requirement specifications?
Non-functional	Accessibility	The system must consider features suitable for disabilities and other users.	Test some accessibility features based on the requirements.	Test and verify all the accessibility features based on the requirements.	Is it accessible for some needs?	Is it accessible for different users?
Non-functional	Usability testing	The ease with which user can learn, operate, prepare inputs and outputs through interaction with the system.	Test the ease of use of the particular feature.	Test the ease of use of the entire features.	Is the system hard to navigate?	Does the system meet user needs?
Non-functional	Stress testing	Testing the system if it can handle extreme workloads like high traffic or data processing.	Join a few members to pay the bill at the same time.	Join as many members as possible to navigate through the system and pay the bill at the same time.	Does the system take longer to confirm payment?	Does the system confirm payment in less than 2 second?

Figure 40: Table showing evaluation plan for during and after development

Here are some example test-cases for after development:

TC ID	Specifications	Steps to Execute	Expected Result	Status
1	Check the error message when the input is invalid for mandatory fields	1. Select add new card 2. Enter letter in card number field 3. Click on submit button	It should display error message for invalid input	Failed
2	Check the sign in page	1. Do not fill any information 2. Click on sign in button	It should display the validation message for all the required fields	Failed
3	Check the sign-up page	1. Fill up all the required input fields 2. Click on sign up button	It should navigate user to the login page	Passed

Figure 41: Example of a test case

Performance Evaluation

For this project, the group targeted for an android bill splitting application. Although avid users of Android applications, we soon realised that our knowledge about the building aspect was scarce. We have then researched the market and we created our first survey. On the first survey, we rushed things and ended up releasing a survey deprived of the necessary support to justify the methods chosen. For this reason, the first survey ended up not being as useful as intended with poor productivity for the project. This prompted us on a second survey being created a few days later, this time following the expected methods and rules to obtain a successful survey that would be reflected in our design. Where we failed on the first survey, we succeed on the second with options as colour scheme and layout being presented.

On Design, we used the data composed by research and surveys and applied it into Use-Cases. This step involved continuous research on which methods to apply to accomplish our goals. Although we found ourselves somewhat unsure about what was the right path to choose, the feedback offered by the supervision definitely facilitated to recentre our efforts on the best pratiques intended on Design.

Technical and Functional Specifications combined all the previous steps into what the foundation of the entire application will be. We analysed the scope and defined its needs and limits, what the application will do or not do; a bird's-eye view offered a full system sight, making it clear what functional and non-functional requirements were needed. At this point, the group's knowledge of the application was fully present and structured; prototyping followed naturally.

Within prototyping, opinion divergences led to deeper research on the matter, and therefore a delay at the start of the next phases. Initially, we had a

misunderstanding about the difference between low and high-fidelity prototypes and the low fidelity wireframes helped us to decide the application's layout, navigation, user interface and features placement. It also helped us to design a high-fidelity prototype, producing an interactive representation of the user interface. This allowed us to get detailed feedback on the design and functionality of the application. Designing the high-fidelity prototype took us a much longer time than expected as we had to make it interactive, include relevant colour scheme and icons for usability, accessibility, and ease of navigation. We conducted an online survey based on the prototype to test our product in terms of usability, accessibility, and functionality. The survey allowed us to improve our product's overall functionality through participants feedback, which resulted in improvements linked directly to the survey results.

Throughout all these weeks and steps, three key aspects were present: communication, productivity, and time management.

Communication

Our group consisted of members that, overall, were complete strangers with a common goal. Despite this and the necessity to only work online, our communication has been improving over time, as members felt more confident as they developed skills to communicate properly and accordingly. We have used mainly Microsoft Teams as our communication tool and as members have learned to better hear and express themselves to the group, outcomes improved. Better communication has increased efficiency and consistency. Two months later there is still room for improvement, but we have learned the importance of good communication in order to get the best results.

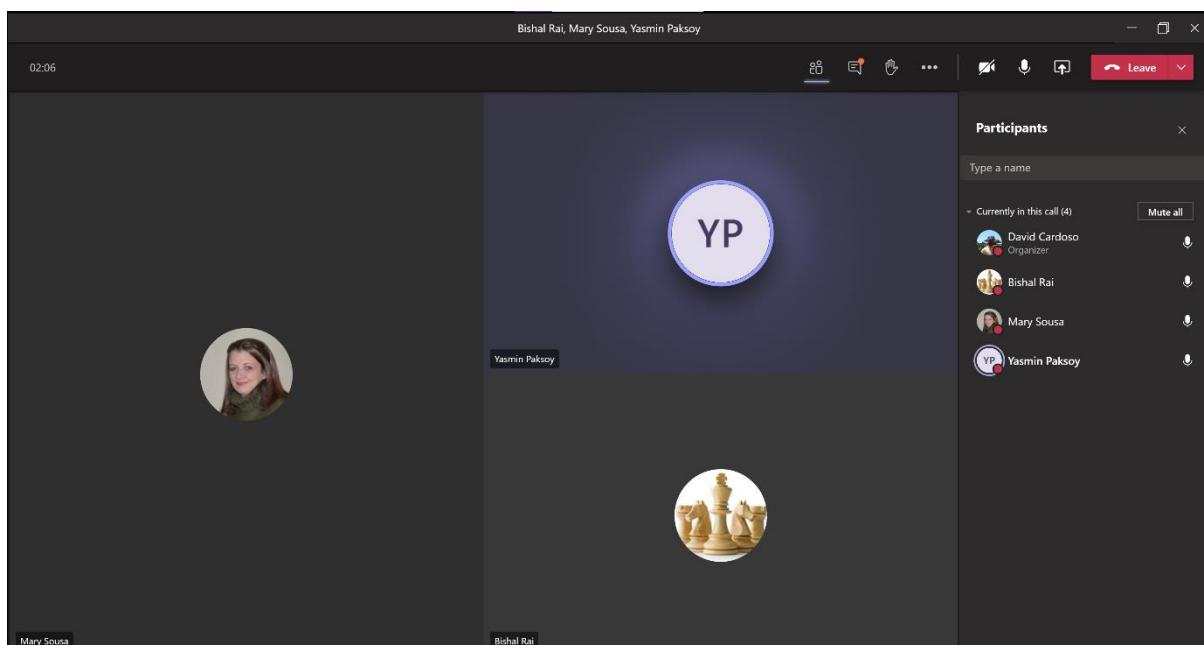


Figure 42: group meeting on Teams

Productivity

As one of the main elements on the creation of a viable product, productivity directly affects the success of a project. For that, we have utilized different tools, such as Notion, that have provided the necessary features to establish and increase productivity. Due to the use of these management tools, along with proper time management, productivity has originated a constant high flow of work which allowed us to see improvements in a recurrent setting.

This constant flow of work produced has been kept on Notion for easy access by the entire group and is also as a backup measure since is saved in the cloud. In the following example, all the documents with research and/or data created by members are accessible to all to be examined and edited.

Documents	Date	Participants
UML	Oct 28, 2020 12:52 AM	(B)D
Questionnaire proposal v1	Oct 28, 2020 12:55 PM	D
Use Cases V1	Oct 28, 2020 9:20 PM	(Y)M
Concept Proposal	Oct 28, 2020 9:33 PM	(Y)MBD
Stakeholders & Requirements	Oct 28, 2020 9:33 PM	(Y)MBD
Gantt Chart	Oct 28, 2020 9:33 PM	(Y)D
Market Research: Restaurant Demographics & Secondary Survey	Oct 30, 2020 3:04 PM	Y
Use Cases V2	Nov 2, 2020 2:35 PM	Y
User Journey Mapping	Nov 2, 2020 6:22 PM	Y
Market Research: Existing apps and target audiences	Nov 3, 2020 4:00 PM	M
Possible names	Nov 4, 2020 10:07 AM	M
Sequence Diagram	Nov 4, 2020 2:33 PM	(Y)B
Normalization	Nov 4, 2020 5:36 PM	D
Wireframes	Nov 4, 2020 9:19 PM	B
Marking Rubric	Nov 5, 2020 4:01 PM	
High Fidelity Prototype: Adobe XD	Nov 9, 2020 7:09 PM	Y
Contrast colour research	Nov 15, 2020 2:50 PM	M
Accessibility fonts and sizes	Nov 15, 2020 3:09 PM	D
Project Management Methodologies Research	Nov 15, 2020 10:18 PM	Y
Testing	Nov 26, 2020 3:53 PM	

Figure 43: Group meeting on Teams

Combined with Notion, we used a Gantt Chart. The visual overview of the project, milestones to be done versus milestones concluded, the time spent versus time available and the reflection of members performance on each milestone provided a major management real-time evaluation. We have used this tool over time with regular updates. Nevertheless, the use of the Gantt chart could have produced better results had we include a more thorough list of milestones in it. Instead, we used general condensed milestones which, when delayed beyond the scheduled, made it harder to identify exactly the task and the reason behind the delay. We only noticed this problem nearly by the end of the part of the project and we will definitely change and do different on the second part of this project.

Overall, our group work methodology was based on Agile meaning that we have done simultaneous and independent work, translated into a continuous workflow. By applying Agile methodology, we have maximized efficiency within each

member since the use time have not being restricted by other member or task. Even when tasks relied on two members, the remaining two members were assigned a similar task where both members could work together.

Lastly, teamwork also enhanced efficiency for together we have tackled and managed issues that individually would require more time and have a bigger dimension in the overview of this project. By sharing a common goal, our group kept a constant covet on the objective, consequently increasing productivity.

Time management

Along with the two previous aspects, time management is crucial to deliver a project with all the technicalities expected, within the expected time.

To better keep track of our time, we scheduled weekly group meeting every Tuesday at 4pm. We have used these meetings to track milestones and tasks, provide feedback on issues and queue the next steps and days. Combined with our weekly group meetings, we also had a weekly supervision meeting where we had showed progress and received feedback. This tight control, aided by the Gantt Chart and the ability to check how far we were compared to the scheduled plan, converged our efforts into tangible results. Each member has a time record table where the time spent is breakdown into individual and meetings times as shown on the following images:

Bishal rai + Add a view							
Week	# Time Spent (hours)	# Time Spent in Me...	Σ Individual & Meeti...	# Weekly Goal	Σ Time Left	Σ Goal Reached	
Week 2	2	2	4	7.33	3.33	No	
Week 3	4	2	6	7.33	1.33	No	
Week 4	3	2	5	7.33	2.33	No	
Week 5	3	2	5	7.33	2.33	No	
Week 6	2	2	4	7.33	3.33	No	
Week 7	3	2	5	7.33	2.33	No	
Week 8	2	3	5	7.33	2.33	No	
Week 9	3	3	6	7.33	1.33	No	
Week 10	3	2	5	7.33	2.33	No	

Figure 44: Bishal Rai time record table

David + Add a view							
Week	# Individual Time Spent ...	# Time spent in meetings	Σ Individual & Meetings Total	# Weekly Goal	Σ Time Left	Σ Goal Reached	
Week 2 - 12/10 to 18/10	1	2	3	7.33	4.33	No	
Week 3 - 19/10 to 25/10	4	1.5	5.5	7.33	1.83	No	
Week 4 - 26/10 to 01/11	3	3.5	6.5	7.33	0.83	No	
Week 5 - 02/11 to 08/11 OPEN	2	1.5	3.5	7.33	3.83	No	
Week 6 - 09/11 to 15/11	3	6.5	9.5	7.33	-2.17	Yes	
Week 7 - 16/11 to 22/11	3	3.5	6.5	7.33	0.83	No	
Week 8 - 23/11 to 29/11	10	3.5	13.5	7.33	-6.17	Yes	
Week 9 - 30/11 to 06/12	4	6	10	7.33	-2.67	Yes	
Week 10 - 07/12 to 11/12	12	2	14	7.33	-6.67	Yes	

Figure 45: David Cardoso time record table

Members: brai001, dcard001, msous001, ypaks001

Mary + Add a view

Aa Week	# Time Spent (hours)	# Time Spent in Meetings	Σ Individual & Meetings T...	# Weekly Goal	Σ Time Left	Σ Goal Reached	+
Week 2	3	1	4	7.33	3.33	No	
Week 3	3	2.5	5.5	7.33	1.83	No	
Week 4	2.5	2.5	5	7.33	2.33	No	
Week 5	3	3	6	7.33	1.33	No	
Week 6	4	3	7	7.33	0.33	No	
Week 7	3	3.5	6.5	7.33	0.83	No	
Week 8	3	3	6	7.33	1.33	No	
Week 9	3	6	9	7.33	-1.67	Yes	
Week 10	5	2	7	7.33	0.33	No	

Figure 46: Mary Sousa time record table

Yasmin + Add a view

Aa Week	# Time Spent (hours)	# Time Spent in Meetings	Σ Individual & Meetings T...	# Weekly Goal	Σ Time Left	Σ Goal Reached	N
Week 2	4	1	5	7.33	2.33	No	
Week 3	2	2.5	4.5	7.33	2.83	No	
Week 4	3.5	1.5	5	7.33	2.33	No	
Week 5	10	3.5	13.5	7.33	-6.17	Yes	
Week 6	7	3.5	10.5	7.33	-3.17	Yes	
Week 7	4	3.5	7.5	7.33	-0.17	Yes	
Week 8	5	1	6	7.33	1.33	No	
Week 9	7	6	13	7.33	-5.67	Yes	
Week 10	12	2	14	7.33	-6.67	Yes	

Figure 47: Yasmin Paksoy time record table

Moreover, we also have a time record table with all the group meeting and all the meeting with supervision:

↗ Week	Aa Name	# Time Spent (hours)
Week 2	↗ M1: Concept Discussion	1
Week 3	↗ M2: Concept & Stakeholders Agreements	2.5
Week 3	↗ n01: Concept Intro	1
Week 4	↗ M3: UML & Use-Cases	1.5
Week 4	↗ n02: UML, Use-Cases and Questionnaire	1
Week 5	↗ M4: Concept & Diagrams Review	1.5
Week 5	↗ n03: Prototyping	2
Week 5	↗ M5: Low Fidelity Prototypes	1.2
Week 6	↗ M6: Prototypes & Report	2
Week 6	↗ n04: XD Prototype	1.5
Week 6	↗ M7: Accessibility, colour schemes and survey 2.0	3
Week 7	↗ n06: Testing Plans	0.45
Week 7	↗ M8: Finalising Prototype	3
Week 8	↗ M9: Testing Feedback	1
Week 9	↗ n07: Report	0.5
Week 9	↗ M10: Report	1.5
Calculate ↴		
		COUNT 18
		SUM 30.65

Figure 48: Group meeting time record table

Though the analysis of this data, it is clear that our time has not been efficient and organized as expected. In 36 possible instances to spend 7.33 hours working on the project as the weekly goal, we only reached it 10 times, which translate into a 28% success rate. There are several possible reasons for this such as agenda overload, complex task taking longer than expected or mere neglection of the timetable. Nevertheless, the group has been able to catch up at the expenses of work overload and probably a decrease in the quality of the result but overall, we have been able to keep a stable workflow.

Conclusion

All things considered, the group has used a distinct range of methodologies and tools to produce a feasible client-based product. Our market research portrayed the user needs, existing applications, and their lacking features. In conjunction with this, our surveys' over 30 results extended our comprehension of what would be the foundation of the application. While Design established the skeleton, Technical & Functional Specification outlined the correlations between pages, features and execution. Prototyping generated a beta model, subjected to the exhaustive scrutiny of 29 users' tests, which resulted in a very refined application. We are now confident that this project has reached adequate maturity to start the

Members: brai001, dcard001, msous001, ypaks001

next steps, where we look forward to producing the relevant technologies and building the client-based product.

Glossary

Agile: Used for describing ways of planning and doing work in which it is understood that making changes as they are needed is an important part of the job [35].

Android: An operating system used mainly for mobile devices that you control by touching the screen [36].

Android Studio: Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development [37].

API: Abbreviation for application programming interface: a set of rules that defines how a software program can request and receive information from other software, usually a website [38].

Fidelity: Design fidelity refers to the level of details and functionality built into a prototype [39].

Hybrid: An application that combines elements of both native and web applications [40].

Java: A cloud-based, object-oriented programming language that is designed to have as few implementation dependencies as possible [41].

Kanban: The Kanban Method is a means to design, manage, and improve flow systems for knowledge work. The method also allows organizations to start with their existing workflow and drive evolutionary change [42].

Minimum viable product: A minimum viable product is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development [43].

MongoDB: MongoDB is a document database which stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time [44].

Native: Designed for or built into a given system, especially denoting the language associated with a given processor, computer, or compiler, and programs written in it [45].

PAN: Primary Account Number, also known as a credit card number [12].

PCI DSS: Payment Card Industry Data Security Standard, a set of industry standards for credit card processors. Compliance is required in order to accept credit card payments [12].

Scrum: Scrum is a process framework used to manage product development and other knowledge work. Scrum is empirical in that it provides a means for teams to establish a hypothesis of how they think something works, try it out, reflect on the experience, and make the appropriate adjustments [46].

Swift: Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. and the open-source community, first released in 2014 [47].

Token: A secret provided by a security system for authentication and verification purposes [12].

UML: Unified Modelling Language (UML) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system [48].

References

- [1] Splitwise. 2019. Privacy Policy. [online] Available at: <https://www.splitwise.com/privacy#:~:text=We share some of your,email address%2C and user avatar> [Accessed 3 November 2020]
- [2] Paypal.com. n.d. How Do-I-Send-Money. [online] Available at: [https://www.paypal.com/us/smarthelp/article/how-do-i-send-money-faq1684#:~:text=With PayPal%2C you can send,email address or mobile number.\)](https://www.paypal.com/us/smarthelp/article/how-do-i-send-money-faq1684#:~:text=With PayPal%2C you can send,email address or mobile number.](https://www.paypal.com/us/smarthelp/article/how-do-i-send-money-faq1684#:~:text=With PayPal%2C you can send,email address or mobile number.)) [Accessed 3 November 2020]
- [3] Tab. 2020. Tab. [online] Available at: <https://www.tabapp.co/> [Accessed 3 November 2020]
- [4] Splitwise. 2020. Get Splitwise Pro!. [online] Available at: [https://www.splitwise.com/subscriptions/new#:~:text=Receipt scanning,will automatically scan its contents.\)](https://www.splitwise.com/subscriptions/new#:~:text=Receipt scanning,will automatically scan its contents.](https://www.splitwise.com/subscriptions/new#:~:text=Receipt scanning,will automatically scan its contents.)) [Accessed 3 November 2020]
- [5] Play.google.com. 2020. [online] Available at: https://play.google.com/store/apps/details?id=com.Splitwise.SplitwiseMobile&hl=en_US&gl=US [Accessed 3 November 2020]
- [6] Play.google.com. n.d. [online] Available at: https://play.google.com/store/apps/details?id=com.bring10.tab&hl=en_US&gl=US&showAllReviews=true [Accessed 15 November 2020]
- [7] Gaille, B., 2016. 38 Staggering Restaurant Demographics. [online] BrandonGaille.com. Available at: <https://brandongaille.com/38-staggering-restaurant-demographics/> [Accessed 3 November 2020]
- [8] Dobronte, A., 2017. Likert Scales vs. Slider Scales in commercial market research. CheckMarket. Available at: <https://www.checkmarket.com/blog/likert-scales-slider-scales/> [Accessed 8 December 2020]
- [9] Preston, C.C. and Colman, A.M. (2000). Optimal number of response categories in rating scales: reliability, validity, discriminating power, and respondent preferences. *Acta Psychologica*, 104(1), pp.1–15. [Accessed 8 December 2020]

- [10] Chapple, M. (2020). *Database Normalization Basics*. Available: <https://www.lifewire.com/database-normalization-basics-1019735>. [Accessed 14 November 2020]
- [11] Ian. (2017). *What is Normalization?*. Available: <https://database.guide/what-is-normalization/>. [Accessed 4 November 2020]
- [12] GDPR Personal Data. (2020) Available at: <https://gdpr-info.eu/issues/personal-data/> [Accessed 10 December 2020]
- [13] MongoDB. 2020. Mongodb Vs MySql. [online] Available at: [https://www.mongodb.com/compare/mongodb-mysql#:~:text=MySQL%20is%20a%20relational%20database,\(SQL\)%20for%20database%20access.&text=MongoDB%20is%20a%20NoSQL%20database,data%20as%20JSON-like%20documents](https://www.mongodb.com/compare/mongodb-mysql#:~:text=MySQL%20is%20a%20relational%20database,(SQL)%20for%20database%20access.&text=MongoDB%20is%20a%20NoSQL%20database,data%20as%20JSON-like%20documents). [Accessed 10 December 2020].
- [14] MongoDB. 2020. Free Cloud Database. [online] Available at: <https://www.mongodb.com/free-cloud-database> [Accessed 10 December 2020]
- [15] Dipina Damodaran, B., Salim, S. and Vargese, S.M., 2016. Performance evaluation of MySQL and MongoDB databases. International Journal on Cybernetics & Informatics (IJCI) Vol, 5.
- [16] RapidAPI Staff. (2020). *The Top 9 Payment APIs To Manage Your Payments*. Available: Access Control – Systems for permitting only authorized users to view and/or change information. API – Application Programming Interface, a system of protocols used for accessing computing res. [Accessed 9 December 2020]
- [17] Developers. Google. (2020). *Introduction to the Payment Request API*. Available: <https://developers.google.com/web/ilt/pwa/introduction-to-the-payment-request-api>. [Accessed 9 December 2020]
- [18] Google Developers. (2020). *Google Pay for Payments Android*. Available: <https://developers.google.com/pay/api/android/overview>. [Accessed 9 December 2020]
- [19] Zhang, Xi. (2020). *What's new in CameraX*. Available: <https://medium.com/androiddevelopers/whats-new-in-camerax-fb8568d6ddc>. [Accessed 9 December 2020]
- [20] Anon, Agile Vs Waterfall: Know the Difference Between Methodologies. Guru99. Available at: <https://www.guru99.com/waterfall-vs-agile.html> [Accessed 1 December 2020]
- [21] Anon, 2020. 5 benefits of agile project management you must know about. Apiumhub. Available at: <https://apiumhub.com/tech-blog-barcelona/benefits-of-agile-project-management/> [Accessed December 11, 2020].

- [22] Rehkopf, M., Kanban vs Scrum. Atlassian. Available at: <https://www.atlassian.com/agile/kanban/kanban-vs-scrum> [Accessed 11 December 2020]
- [23] Wong, E., 2020. Principle Of Consistency And Standards In User Interface Design. [online] The Interaction Design Foundation. Available at: <https://www.interaction-design.org/literature/article/principle-of-consistency-and-standards-in-user-interface-design> [Accessed 1 December 2020]
- [24] The Redbooth Team, 2015. Ios Hamburger Menu: Why We Banished One From Our App. [online] Redbooth. Available at: https://redbooth.com/blog/hamburger-menu-iphone-app?utm_campaign=iOS_Dev_Weekly_Issue_181&utm_medium=email&utm_source=iOS%2BDev%2BWeekly [Accessed 1 December 2020]
- [25] Hauff, A., 2018. Color Psychology In Marketing: The Complete Guide [Free Download]. [online] CoSchedule Blog. Available at: <https://coschedule.com/blog/color-psychology-marketing/> [Accessed 7 November 2020]
- [26] World Health Organization. (2020). *Disability and health*. Available: <https://www.who.int/news-room/fact-sheets/detail/disability-and-health>. [Accessed 8 December 2020]
- [27] Hassell, J. (2018). *The importance of text accessibility: how IBM's Content Clarifier shows us what we've forgotten*. Available: <https://www.hassellinclusion.com/blog/importance-text-accessibility/>. [Accessed 8 December 2020]
- [28] W3C. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. Available: <https://www.w3.org/TR/WCAG21/>. [Accessed 8 December 2020]
- [29] Ford Williams, G. (2020). A Guide to Understanding What Makes a Typeface Accessible. Available: <https://uxdesign.cc/a-guide-to-understanding-what-makes-a-typeface-accessible-and-how-to-make-informed-decisions-9e5c0b9040a0>. [Accessed 8 December 2020]
- [30] Dr Tedros Adhanom Ghebreyesus. (2020). blindness-and-visual-impairment. |World Health Organization [online] Available: <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>. [Accessed 3 December 2020]
- [31] National Literacy Trust. n.d. Adult Literacy | National Literacy Trust. [online] Available at: <https://literacytrust.org.uk/parents-and-families/adult-literacy/> [Accessed 4 December 2020]

- [32] Geerlings, C., 2019. How To Create An Accessible App (And Why You Should). [online] Medium. Available at: <https://medium.com/oberonamsterdam/how-to-create-an-accessible-app-and-why-you-should-5493f41f8bdb> [Accessed 4 December 2020]
- [33] Chen, K., Savage, A., Chourasia, A., Wiegmann, D. and Sesto, M., 2013. Touch screen performance by individuals with and without motor control disabilities. *Applied Ergonomics*, 44(2), pp.297-302. [Accessed 4 December 2020]
- [34] Gebicz, M. (2020). *What is a Gantt Chart?*. Available: <https://www.atlassian.com/agile/project-management/gantt-chart>. Last accessed [Accessed 4 December 2020]
- [35] Anon, AGILE: meaning in the Cambridge English Dictionary. Cambridge Dictionary. Available at: <https://dictionary.cambridge.org/dictionary/english/agile> [Accessed 6 December 2020]
- [36] Anon, android: meaning in the Cambridge English Dictionary. Cambridge Dictionary. Available at: <https://dictionary.cambridge.org/dictionary/english/android> [Accessed 6 December 2020]
- [37] Anon, Meet Android Studio: Android Developers. Android Developers. Available at: <https://developer.android.com/studio/intro> [Accessed 6 December 2020]
- [38] Anon, API: meaning in the Cambridge English Dictionary. Cambridge Dictionary. Available at: <https://dictionary.cambridge.org/dictionary/english/api> [Accessed 6 December 2020]
- [39] Pacheco, M., 2014. Understanding Design Fidelity for Creating a Great Product Experience. Cantina. Available at: <https://cantina.co/understanding-design-fidelity-for-creating-a-great-product-experience/> [Accessed 6 December 2020]
- [40] Rouse, M., 2019. What is a Hybrid Application (Hybrid App)? SearchSoftwareQuality. Available at: <https://searchsoftwarequality.techtarget.com/definition/hybrid-application-hybrid-app> [Accessed 6 December 2020]
- [41] Schildt, H., 2019. Java: a beginner's guide, New York: McGraw-Hill Education.
- [42] Anon, 2019. What is Kanban? Agile Alliance. Available at: <https://www.agilealliance.org/glossary/kanban/> [Accessed 6 December 2020].

- [43] Becker, R., What is a Minimum Viable Product (MVP)? - Definition from Techopedia. Techopedia.com. Available at: <https://www.techopedia.com/definition/27809/minimum-viable-product-mvp> [Accessed 6 December 2020].
- [44] Anon, What Is MongoDB? MongoDB. Available at: <https://www.mongodb.com/what-is-mongodb> [Accessed 6 December 2020].
- [45] Anon, Native: Definition of Native by Oxford Dictionary on Lexico.com also meaning of Native. Lexico Dictionaries | English. Available at: <https://www.lexico.com/definition/native> [Accessed 6 December 2020].
- [46] Anon, 2019. Scrum. Agile Alliance. Available at: <https://www.agilealliance.org/glossary/scrum/> [Accessed 6 December 2020].
- [47] Inc., A., Swift. Apple Developer. Available at: <https://developer.apple.com/swift/> [Accessed 6 December 2020].
- [48] Anon, What is Unified Modeling Language (UML)? Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> [Accessed 6 December 2020].

Bibliography

- Ambler, S. (2019). Agile Modeling. UML 2 Class Diagrams: An Agile Introduction. [online] Available at: <http://www.agilemodeling.com/artifacts/classDiagram.htm> [Accessed 20 November 2020]
- Android Developers. (2020). *Build your first app.* Available: <https://developer.android.com/training/basics/firstapp/>. [Accessed 5 December 2020]
- Beuzeval , A. MY MEMORY. (2019). Best Bill Splitting Apps: Easily Split Bills With Friends. [online]. Available at: <https://www.mymemory.co.uk/blog/best-bill-splitting-apps-easily-split-bills-with-friends/> [Accessed 15 Nov. 2020].
- Chao Wang, Wei Duan, Jianzhang Ma and Chenhui Wang, "The research of Android System architecture and application programming," Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, 2011, pp. 785-790, doi: 10.1109/ICCSNT.2011.6182081.
- Cohen, D., Lindvall, M. and Costa, P., 2004. An introduction to agile methods. *Adv. Comput.*, 62(03), pp.1-66.
- Corbalan, L., Fernandez, J., Cuitiño, A., Delia, L., Cáseres, G., Thomas, P. and Pesado, P., 2018, May. Development frameworks for mobile devices: a comparative study about energy consumption. In Proceedings of the 5th International Conference on Mobile Software Engineering and Systems (pp. 191-201).
- D. Ameller, C. Ayala, J. Cabot and X. Franch, "Non-functional Requirements in Architectural Decision Making," in IEEE Software, vol. 30, no. 2, pp. 61-67, March-April 2013, doi: 10.1109/MS.2012.176.
- Gilliam haije, E. (2020). Top 20 Mobile Development Tools: An Overview. [Online]. Available at: <https://mopinion.com/mobile-development-tools-an-overview/> [Accessed 23 November 2020].
- Basart, J.M. and Serra, M., 2013. Engineering ethics beyond engineers' ethics. *Science and engineering ethics*, 19(1), pp.179-187. [Accessed 23 November 2020].
- Hemsworth , M. (2019). Real-Time Payment-Splitting Apps. 20th April 2018. [online] Available at: <https://www.trendhunter.com/trends/rift-pay> [Accessed 14 November 2020].
- Hofmeister, C., Nord, R.L. and Soni, D., 1999, February. Describing software architecture with UML. In Working Conference on Software Architecture (pp. 145-159). Springer, Boston, MA.
- Holla, S. and Katti, M.M., 2012. Android based mobile application development and its security. *International Journal of Computer Trends and Technology*, 3(3), pp.486-490.

Iman, A.M. UX Collective. (2019). Redesigning an app that helps you split a bill — a UX case study. [online]. Available at: <https://uxdesign.cc/splitting-a-bill-at-a-restaurant-4eab00b42795> [Accessed 18 November 2020]

Irby, L. (2020). 6 Best Bill-Splitting Apps of 2020. [online]. Available at: <https://www.thebalance.com/best-bill-splitting-apps-4170968> [Accessed 14 November 2020].

Islam, S., 2014. Systematic literature review: Security challenges of mobile banking and payments system. International Journal of u-and e-Service, Science and Technology, 7(6), pp.107-116.

Joshi, A. et al., 2015. Likert Scale: Explored and Explained. British Journal of Applied Science & Technology, 7(4), pp.396-403.

Kadhiwal, S. and Zulfiqar, A.U.S., 2007. Analysis of mobile payment security measures and different standards. Computer Fraud & Security, 2007(6), pp.12-16.

A. Abernathy, X. Yuan, E. Hill, J. Xu, K. Bryant and K. Williams, "SACH: A tool for assisting Secure Android application development," SoutheastCon 2017, Charlotte, NC, 2017, pp. 1-4, doi: 10.1109/SECON.2017.7925374.

Kamthan, P., 2009. Ethics in software engineering. In Software Applications: Concepts, Methodologies, Tools, and Applications (pp. 2795-2802). IGI Global.

Kantamani, S. (2020). *How to Integrate Google Pay Into Your Existing Android App*. Available: <https://medium.com/better-programming/how-to-integrate-google-pay-into-your-existing-android-app-d75b269cd623>. [Accessed 5 December 2020]

mongodb.com. (2020). what-is-mongodb. [online] Available at: <https://www.mongodb.com/what-is-mongodb> [Accessed 22 November 2020].

Mukherjee, S. and Mondal, I., 2014. Future practicability of Android application development with new Android libraries and frameworks. Int. J. Comput. Sci. Inf. Technol, 5(4), pp.5575-5579.

Pardeshi, G. WhaTech. (2020). Explore the Bill Splitting Apps Market Forecasts to 2026. [Online]. Available at: <https://www.whatech.com/markets-research/it/640159-explore-the-bill-splitting-apps-market-forecasts-to-2026> [Accessed 15 November 2020].

Shahir. Weekly Wall. (2020). Trending 2020 Bill Splitting Apps Market Size, Analysis, and Forecast Report 2020 -2026. [online]. Available at: <https://weeklywall.com/trending-2020-bill-splitting-apps-market-size-analysis-and-forecast-report-2020-2026/> [Accessed 18 November 2020]

Techopedia. (2020). *Normalization*. Available: <https://www.techopedia.com/definition/1221/normalization>. [Accessed 4 November 2020]

Thomson, A.J. and Schmoldt, D.L., 2001. Ethics in computer software design and development. Computers and Electronics in Agriculture, 30(1-3), pp.85-102.

Unjiya, V. (2020). Top 7 Technologies Used For Creating Robust Mobile Apps in 2020. [Online]. [27 November 2020]. Available at: <https://yourstory.com/mystory/top-mobile-app-development-technologies-2020> [Accessed 23 November 2020]

Walker, M., Takayama, L. and Landay, J.A., 2002, September. High-fidelity or low-fidelity, paper or computer? Choosing attributes when testing web prototypes. In Proceedings of the human factors and ergonomics society annual meeting (Vol. 46, No. 5, pp. 661-665). Sage CA: Los Angeles, CA: SAGE Publications.

Watkin, H. (2018). 5 of the Best Cost-Splitting Apps for Travelers. [online]. Available at: <https://toomanyadapters.com/best-cost-splitting-apps/> [Accessed 14 November 2020]

wikipedia.org. (2020). Sequence diagram. [online] Available at: https://en.wikipedia.org/wiki/Sequence_diagram [Accessed 18 November 2020]

Wired. (2020). 7 Apps That Make It Simple to Split the Bill. [online] Available at: <https://www.wired.com/story/expense-bill-splitting-apps/> [Accessed 14 November 2020]

W3C. (2020). *Payment Request API, W3C Candidate Recommendation Snapshot 03 December 2020*. Available: <https://www.w3.org/TR/payment-request/#paymentrequest-interface>. [Accessed 9 December 2020]

Appendices

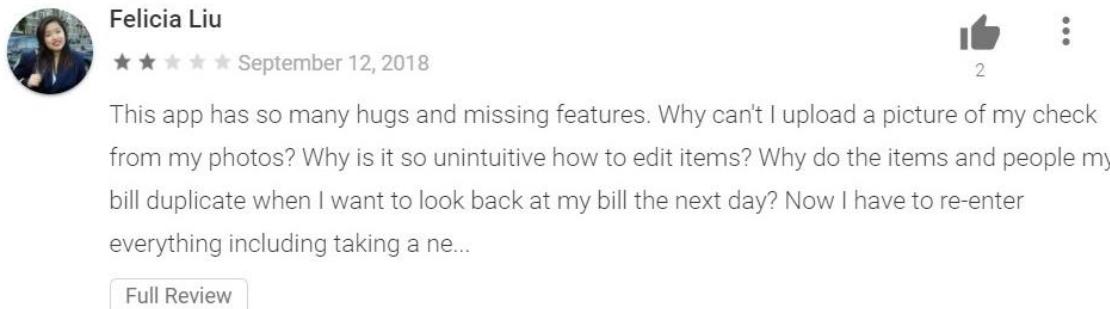


Figure 49: User review on Google Play Store for the application 'Tab'.

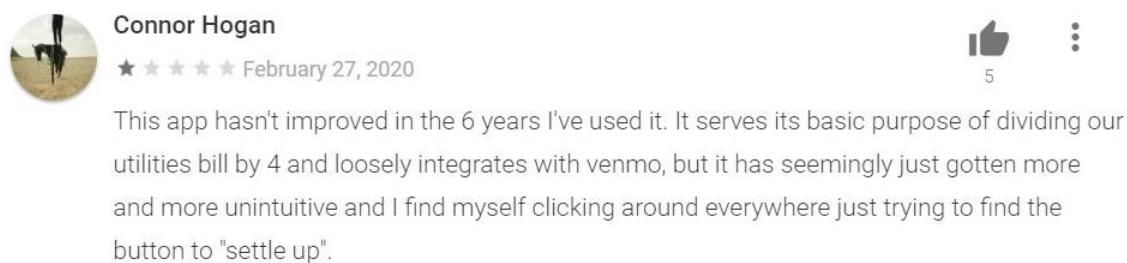


Figure 50: User review on Google play Store for the application 'Splitwise'.

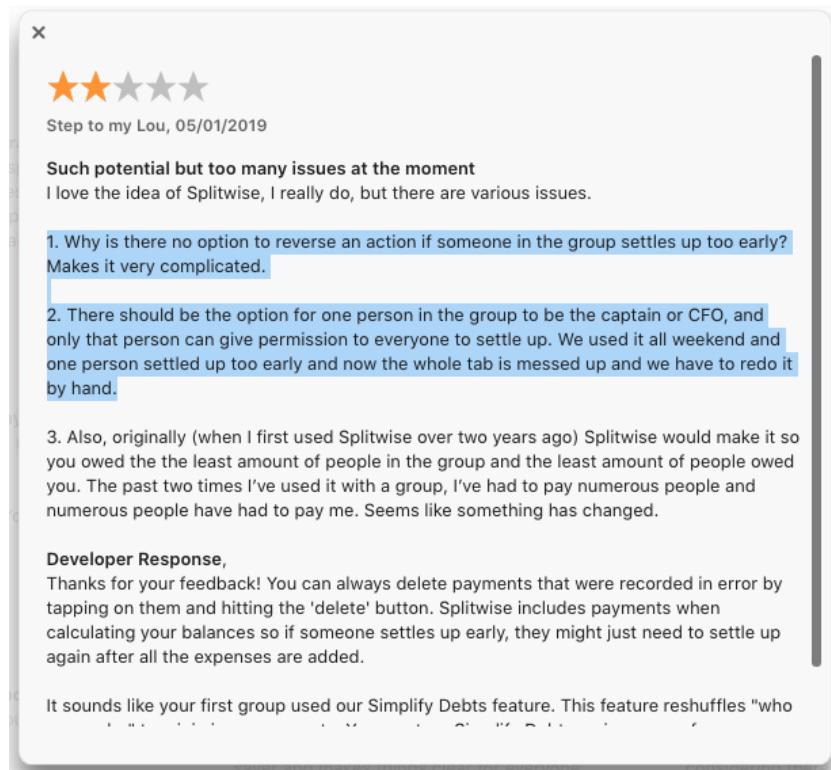


Figure 51: User review on Google play Store for the application 'Splitwise'.

Which age group are you in? *

- < 18
- 18 - 24
- 25 - 34
- 35 - 44
- 45 - 54
- 55 - 64
- 65+

Figure 52: Initial market survey question 1

Which age group are you in?

19 responses

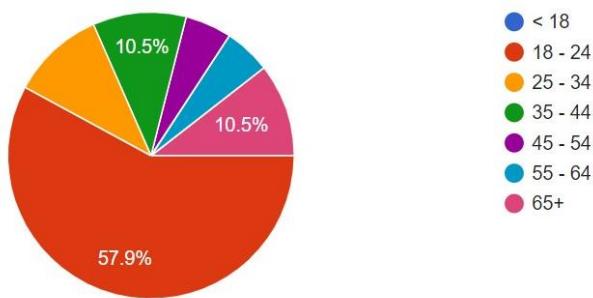


Figure 53: Initial market survey question 1 results

Before COVID-19, how many times per month would you split a bill with others? *

- Rarely
- 1 - 2
- 3 - 4
- 4+

Figure 54: Initial market survey question 2

Before COVID-19, how many times per month would you split a bill with others?
19 responses

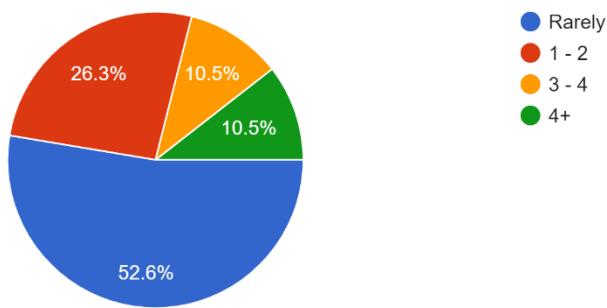


Figure 55: Initial market survey question 2 results

How do you split bills? *

- Cash in hands
- Bank transfer
- PayPal
- Ewallets
- Already use splitting apps (Plates, SplitWise, etc)
- Other: _____

Figure 56: initial market survey question 3

How do you split bills?

19 responses

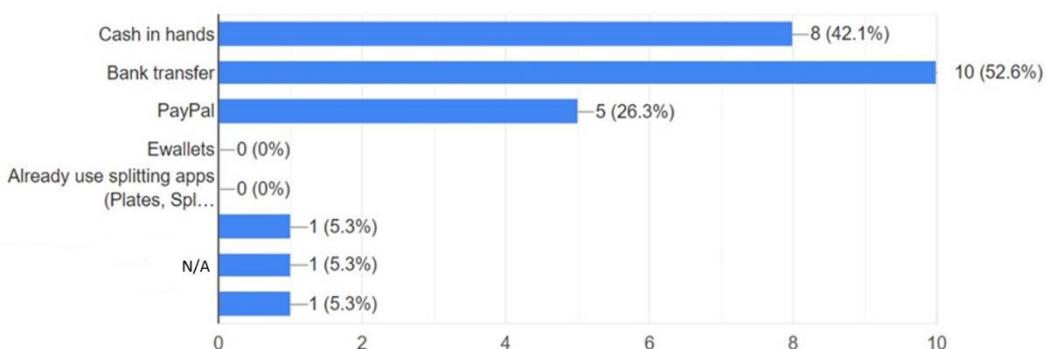


Figure 57: Initial market survey question 3 results

Would you use an app that facilitates splitting a bill among different people? *

For example, sharing a dinner bill or splitting a utility bill

- No
- Maybe
- Yes
- Already use

Figure 58: Initial market survey question 4

Would you use an app that facilitates splitting a bill among different people?

19 responses

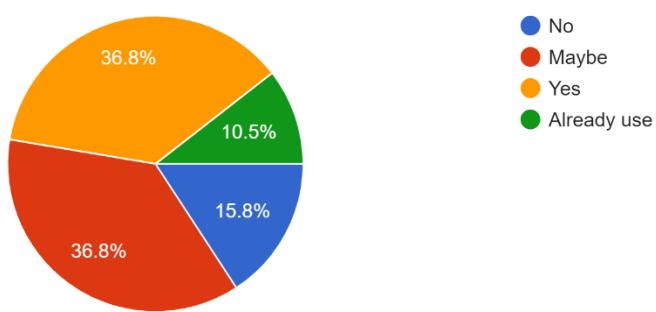


Figure 59: Initial market survey question 4 results

In a cost splitting app, please classify from 0 to 4 how important would be the following features: *

	0 - Irrelevant	1 - Somewhat irrelevant	Not irrelevant or important	3 - Important	4 - Essential
Splitting the bill equally by the number of participants	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Choosing exactly what item you have consumed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Data encryption	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Privacy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spending report	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 60: Initial market survey question 5

In a cost splitting app, please classify from 0 to 4 how important would be the following features:

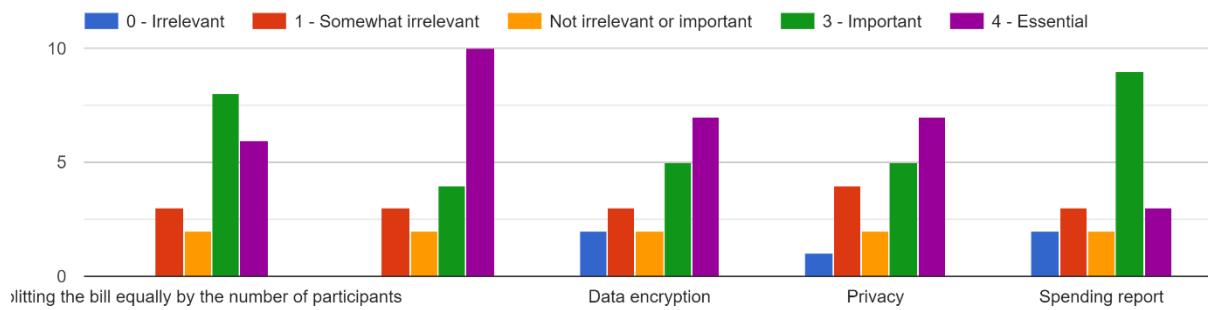


Figure 61: Initial market survey question 5 results

Any feature suggestion not covered above? Please describe bellow.

Your answer

Figure 62: Initial market survey question 6

Any feature suggestion not covered above? Please describe bellow.

7 responses

Let's you know your mo that spendings on food/drinks?

N/A

thank you for your time

Have clear call to action, ease of transaction,

The ability to save payment details so you don't have to type it in each time.

Just flexibility. Some people are happy to split e.g four ways, others may have limited cash and want to pay for only what they eat or drink. Flexibility in the system would be good.

Figure 63: Initial market survey question 6 results

Which age group are you in? *

< 23
 24-40
 41-55
 56-74
 75+

Figure 64: Secondary market survey question 1

Which age group are you in?

33 responses

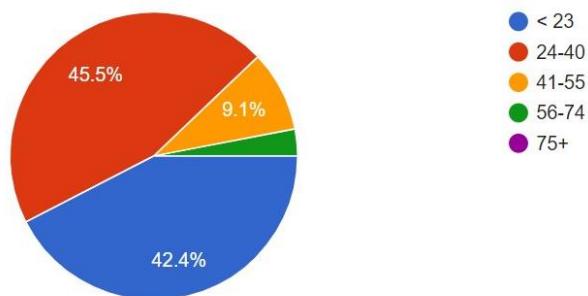


Figure 65: Secondary market survey question 1 results

How often do you go out each month? (Before Covid-19) *

- Never
- 1-4 times (around once a week)
- 5-8 times (around twice a week)
- 9-12 times (around 3 times a week)
- 13+ times (around 4 times a week)

Figure 66: Secondary market survey question 2

How often do you go out each month? (Before Covid-19)

33 responses

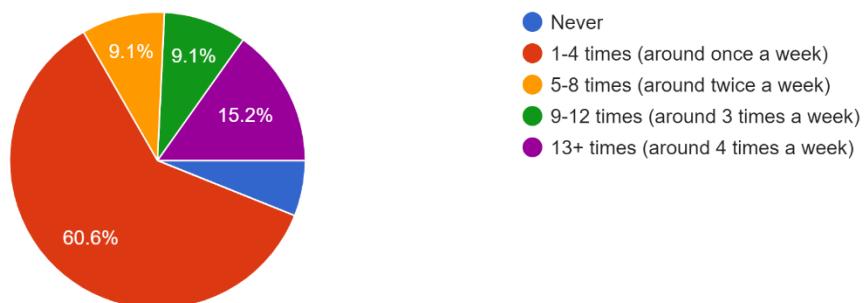


Figure 67: Secondary market survey question 2 results

How often would you split a bill per month? (Before Covid-19) *

- Never
- 1-4 times (around once a week)
- 5-8 times (around twice a week)
- 9-12 times (around 3 times a week)
- 13+ times (around 4 times a week)

Figure 68: Secondary market survey question 3

How often would you split a bill per month? (Before Covid-19)

33 responses

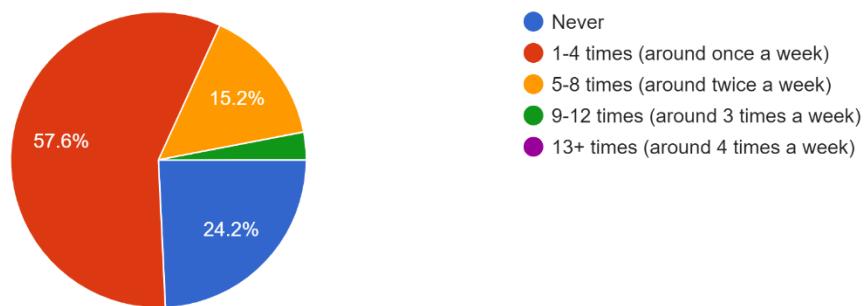


Figure 69: Secondary market survey question 3 results

Which type of bills are you most likely to be splitting?

- Rent
- Utilities, e.g. water, gas, electric bills
- Small subscription services, e.g. Netflix, Spotify
- Events with friends, e.g. going to restaurants, cinemas etc.
- Other: _____

Figure 70: Secondary market survey question 4

Which type of bills are you most likely to be splitting?

32 responses

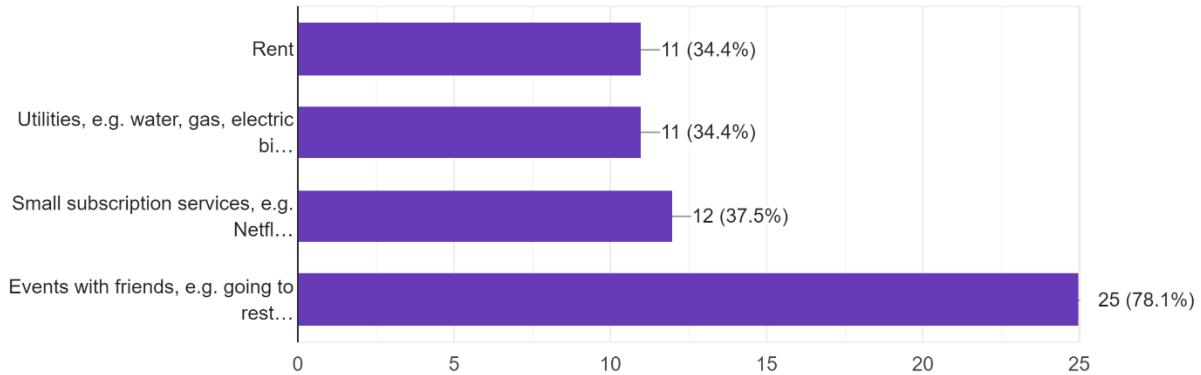


Figure 71: Secondary market survey question 4 results

How do you currently split bills? Select all that apply. *

- Bank transfer
- Cash in hand
- PayPal
- E-Wallets
- Existing bill splitting applications
- I don't split bills
- Other: _____

Figure 72: Secondary market survey question 5

How do you currently split bills? Select all that apply.

33 responses

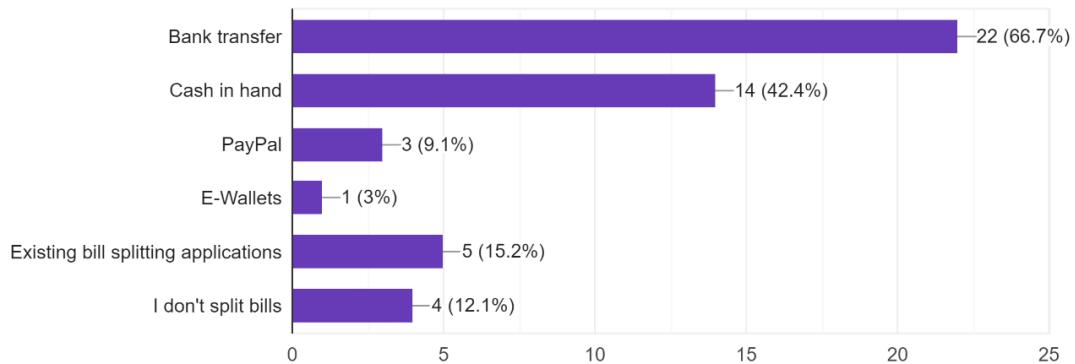


Figure 73: Secondary market survey question 5 results

How interested would you be in a app that facilitates splitting a bill? *

This can be punctual or recurrent. For example, sharing a dinner bill or splitting a utility bill.

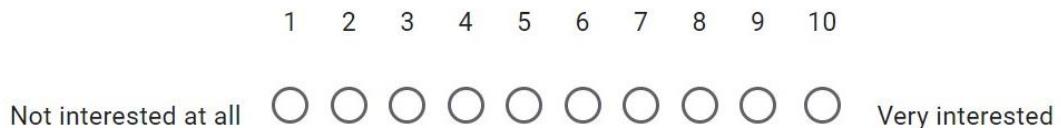


Figure 74: Secondary market survey question 6

How interested would you be in a app that facilitates splitting a bill?

33 responses

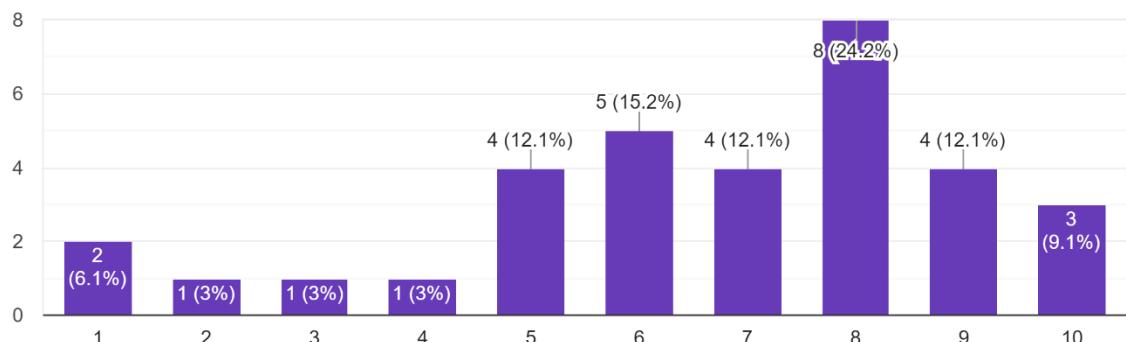


Figure 75: Secondary market survey question 6 results

How important would be the ability to choose to pay for specific items when using such an app?

1 2 3 4 5 6 7 8 9 10

Not Important Very Important

Figure 76: Secondary market survey question 7

How important would be the ability to choose to pay for specific items when using such an app?
33 responses

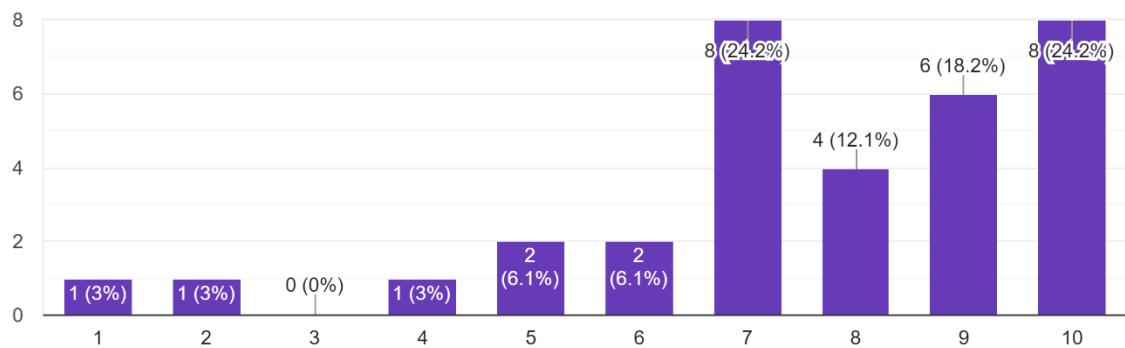


Figure 77: Secondary market survey question 7 results

How important would be the ability to use Google/Apple Pay when using such an app?

1 2 3 4 5 6 7 8 9 10

Not Important Very Important

Figure 78: Secondary market survey question 8

How important would be the ability to use Google/Apple Pay when using such an app?

33 responses

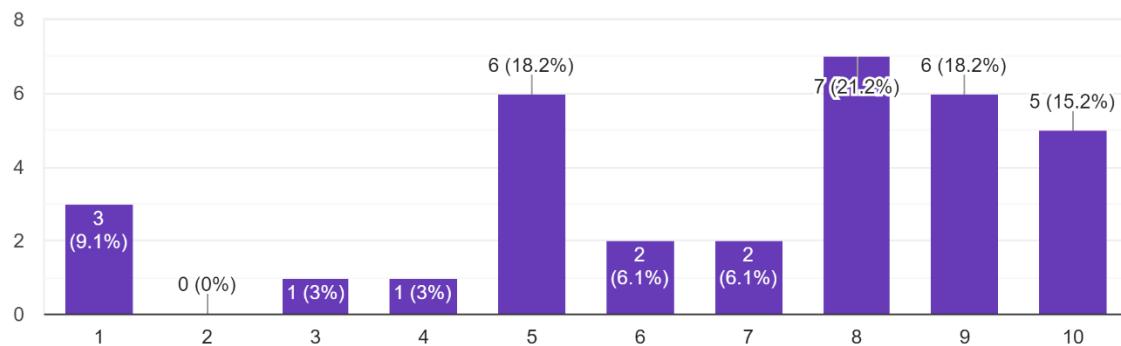


Figure 79: Secondary market survey question 8 results

If there are any other features you'd like to see on a bill splitting application, please leave them below:

Your answer

Figure 80: Secondary market survey question 9

If there are any other features you'd like to see on a bill splitting application, please leave them below:

6 responses

Contacts page

How much it would be along with the service charge added so I don't have to work it out & out on extra myself

profile picture

Help feature that guide users on how the apps work

Figure 81: Secondary market survey question 9 results

Members: brai001, dcard001, msous001, ypaks001

Please select the colour scheme you like the most:

33 responses

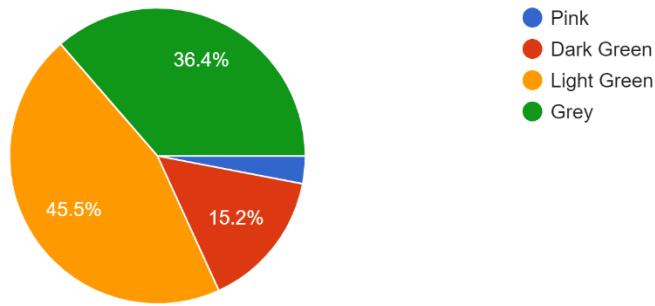


Figure 82: Secondary market survey question 11 results



Figure 83: Original use-case diagram for bill member

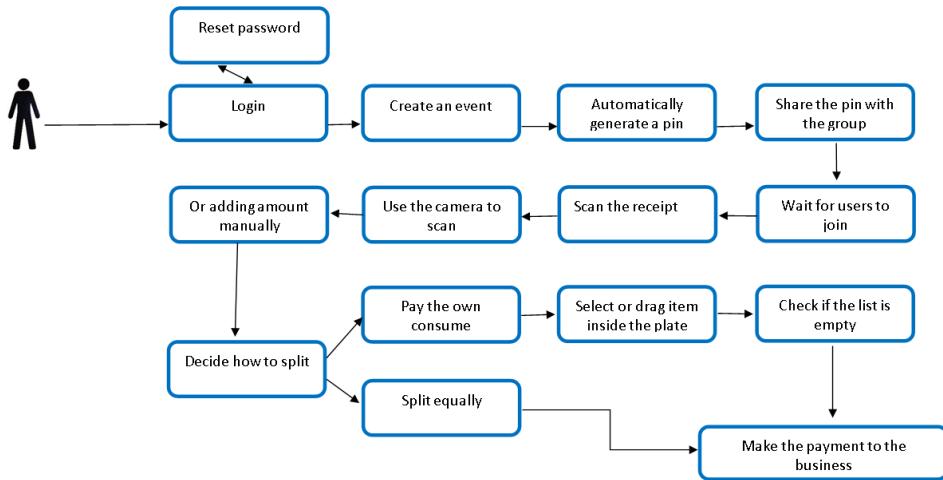


Figure 84: Original use-case diagram for bill creator

	AWARENESS	SERVICE			Loyalty
Timeline	Step 1	Step 2	Step 3	Step 4	Step 5
User Thinking	User is having to split a bill with a group of people and does not want to have to share bank account information with everyone.	Each member of the group needs to be paying a different amount as they have each consumed different items.	User cannot trust that all group members will pay them back once they pay the bill.	User now must figure out how much tip everyone should be paying.	User has now been paid back by all the bill members and is happy with their experience using the application.
Touchpoint	User opens application and goes to create a new bill to make splitting the bill easier on the whole group.	The user selects 'Split by consumption' and scans the bill, allowing group members to select the items they should be paying for.	User selects a pay-by date, allowing the application to make sure bill members do pay them back by a certain date.	User inputs the total amount the group wants to tip into the app, and it is equally divided between the member in the group and added to their totals.	User submits a review on the app store recommending the application to other users.
Channel Experience	Mobile application. User experience is positive; they have an easy way to split their bill without having to spend time sharing bank account information.	Mobile application. User experience is positive; they do not have to work out individual bills like they previously would and have an easy way to calculate individual bills.	Mobile application. User experience is positive; they do not have to worry about whether bill members will pay them back or not.	Mobile application. User experience is positive; they do not have to now spend time figuring out how much everyone should tip.	App Store. User has had a overall positive experience.

Figure 85: Initial user-journey mapping diagram

Members: brai001, dcard001, msous001, ypaks001



Figure 86: Initial version of high-fidelity prototype



Figure 87: Version 1 of complete high-fidelity prototype



Figure 88: Version 2 of high-fidelity prototype

How intuitive was it to know which buttons to click next? *

1 2 3 4 5

Not intuitive Very intuitive

Figure 89: Remote testing survey question 1

How intuitive was it to know which buttons to click next?

19 responses

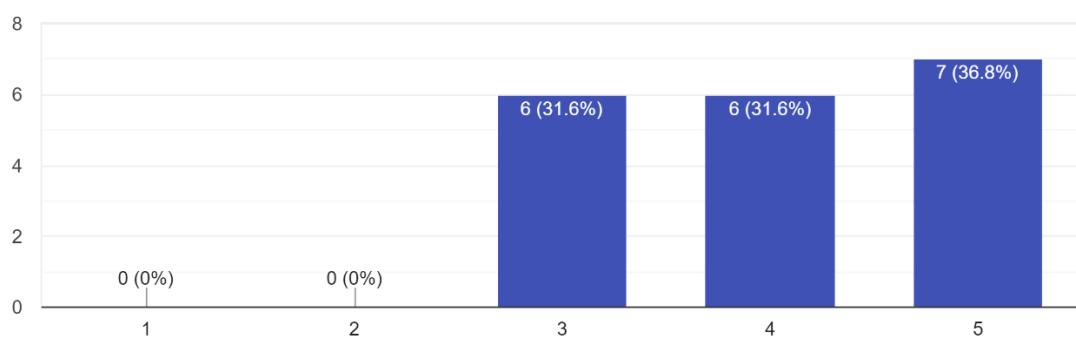


Figure 90: Remote testing survey question 1 results

How easy was the colour scheme to read? *

1 2 3 4 5

Difficult Easy

Figure 91: Remote testing survey question 2

How easy was the colour scheme to read?

19 responses

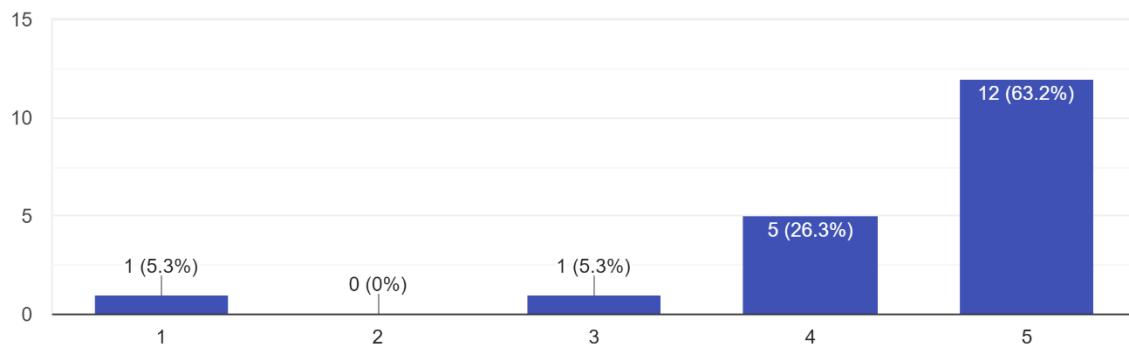


Figure 92: Remote testing survey question 2 results

How clear were the instructions on the app? *

1	2	3	4	5	
Not clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very clear

Figure 93: Remote testing survey question 3

How clear were the instructions on the app?

19 responses

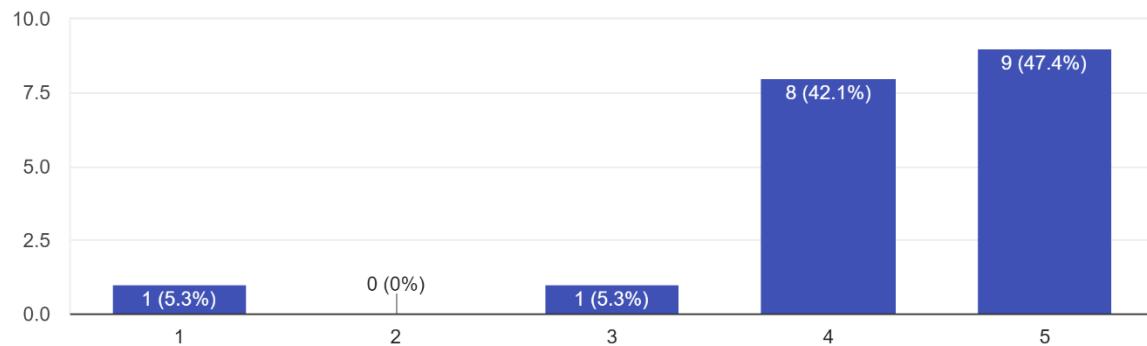


Figure 94: Remote testing survey question 3 results

How relevant were the icons to what they are supposed to be representing? *

1 2 3 4 5

Not relevant Very relevant

Figure 95: Remote testing survey question 4

How relevant were the icons to what they are supposed to be representing?

19 responses

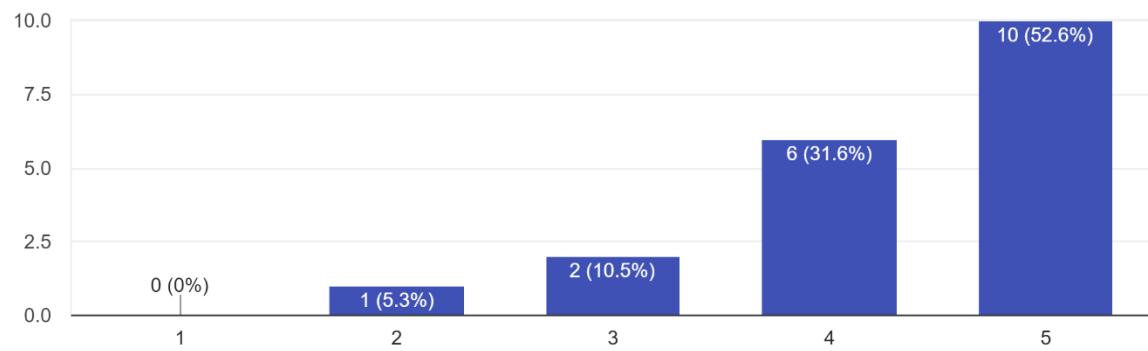


Figure 96: Remote testing survey question 4 results

How helpful were the icons when navigating the app? *

1 2 3 4 5

Not helpful Very helpful

Figure 97: Remote testing survey question 5

How helpful were the icons when navigating the app?

19 responses

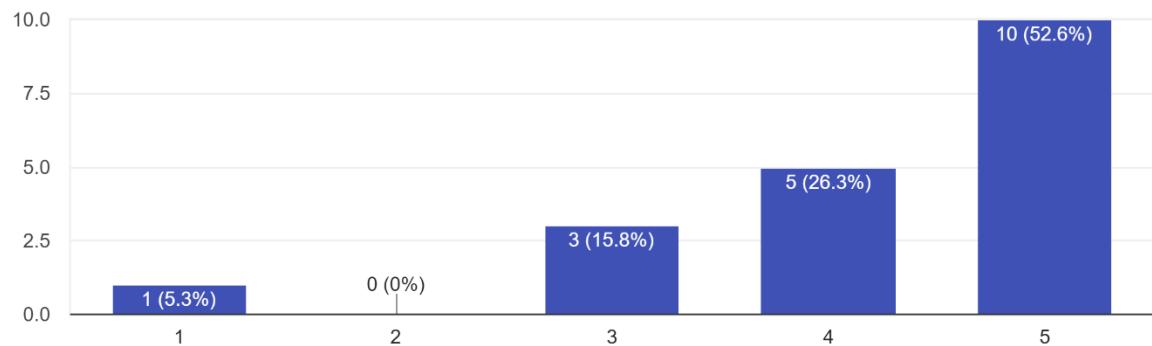


Figure 98: Remote testing survey question 5 results

How easy was it to reach the high contrast option? *

A horizontal scale for Question 6. It consists of five numbered circles from 1 to 5, with "Difficult" at the left end and "Easy" at the right end. The numbers 1, 2, 3, 4, and 5 are positioned above their respective circles.

Figure 99: Remote testing survey question 6

How easy was it to reach the high contrast option?

19 responses

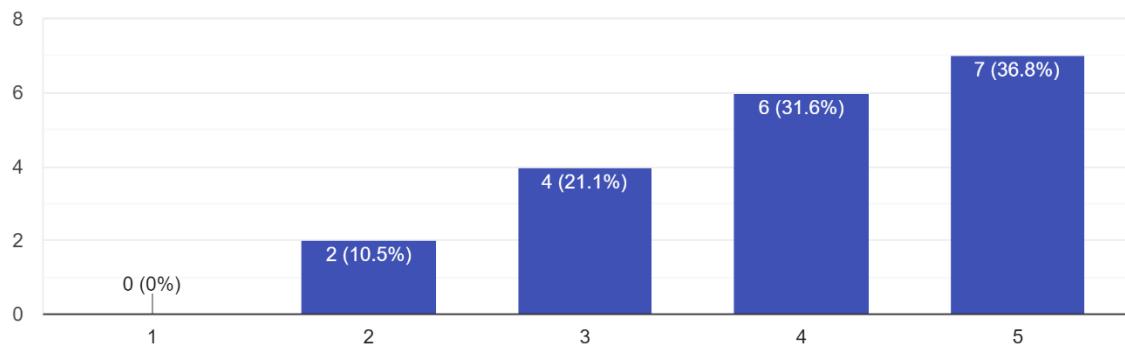


Figure 100: Remote testing survey question 6 results

Please leave any addition feedback below:

Your answer

Figure 101: Remote testing survey feedback box

Amazing app Project Planner

PHASE	BREAKDOWN	PLAN DURATION		ACTUAL DURATION	PERCENT COMPLETE	Plan Duration		Actual Start		% Complete		Actual (beyond plan)															
		OCTOBER	NOVEMBER			DEZEMBER	JANUARY	FEBRUARY	MARCH	APRIL																	
		1	2			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1. Requirements	Initial concept, stakeholder research and requirements UML, questionnaire, Gantt chart	1	1	2	25%																						
2. Design	Planning																										
	Built & test prototypes																										
	Technical Architecture and Specifications																										
3. Development & Testing																											
DURATION OF THE PROJECT		24	1	3	0%																						

Figure 102: Initial version of the Gantt chart

Splitsy App Project



Figure 103: Penultimate version of the Gantt chart