SW Engineering CSC 648/848 Fall 2022

Project: Bill Splitting Web App

FunBill

Team 01

Denyse Luzon	Team Lead, dluzon1@sfsu.edu
Bhagdeep Sandhu	Database Master
Faiyaz Chaudhury	Backend Lead
Kaung Nay Htet (Kevin)	Backend Team
Poornank Purohit	Frontend Lead
Tolby Lam	Frontend Team, GitHub Master

Milestone 1

Milestone/Version	Date
M1V1	09/22/2022
M1V2	10/20/2022

Table of Contents

Back to Table of Contents

Section I: Project Description

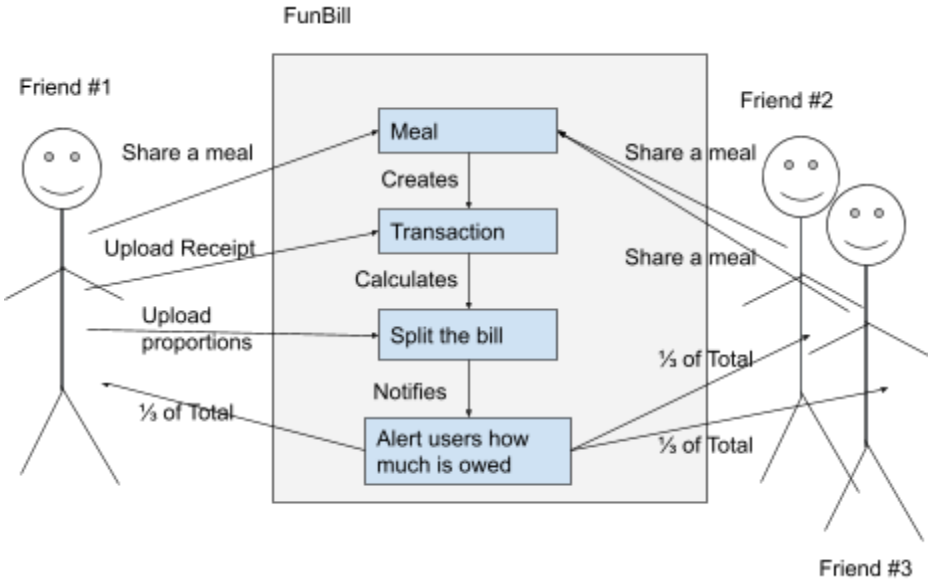
FunBill is a web application that is designed to allow groups of individual users to keep track of and split costs quickly and easily. By grouping costs and proportioning them out to the members, nobody now pays for expenses unfairly. By letting users form groups simply and allowing them to pay in-app, FunBill is an extremely powerful and versatile tool for wherever people need to split bills. FunBill also allows users to add transactions of multiple currencies, which makes cross-continental travel trouble-free and worry-free with its straightforward design principles.

The web application organizes all the transactions so that everyone can see and keep track of how much they owe. Whether you are sharing a Ski trip, splitting rent with roommates, or paying someone back for lunch, FunBill allows users to make life easier. Our web application allows its users to create transactions, request payments from other users, and complete payments all within the app. Users can opt to either create groups for extended periods or pay each other one-on-one through one-time transactions. Users also have the freedom to use different currencies for the transactions needed to be completed within the web application. So whether you are traveling abroad or within your home town, you can know you will be able to pay those you owe. Users can search for transactions by date, users, groups, and/or categories using FunBill's powerful search feature. All to keep track of your history and confirm whether payments have been completed.

Back to Table of Contents

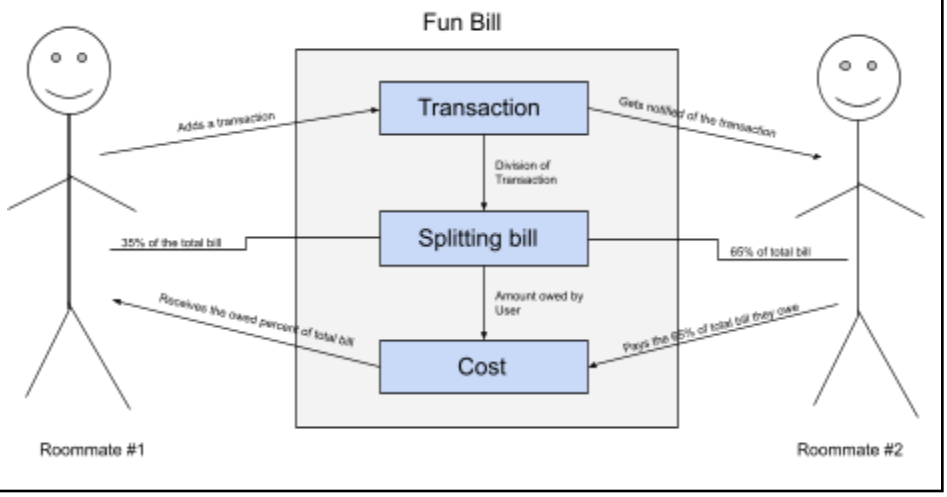
Section II: Use Cases

1. A Group of Friends Goes to a Restaurant

Use Case	A Group of Friends Goes to a Restaurant
Actors	John (User), Dave (User), Sue (User)
Description	John, Dave, and Sue have not seen each other for years and want to reconnect at San Tung. However, at the end of the meal, Sue realizes she forgot her wallet and Dave asks John to pick up the tab. John is annoyed since he knows that they cannot calculate the amount everyone will owe him easily.
Solution	FunBill allows John to easily record the totals from each receipt and split them three ways after the fact. With it, everyone came to a conclusion and left happy.
Diagram	

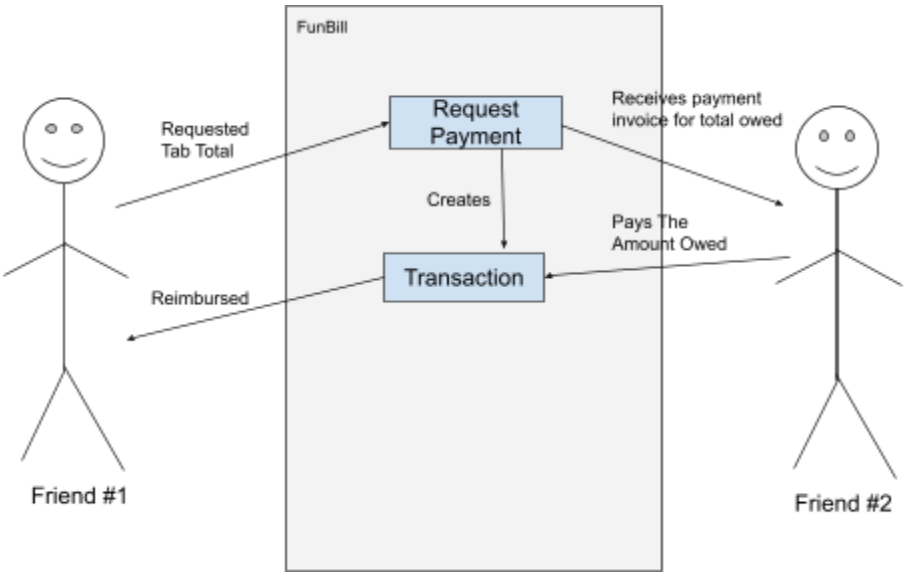
Back to Table of Contents

2. Roommates want to split groceries

Use Case	Roommates want to split groceries
Actors	Alice (User), Stacy (User), FunBill (Web App)
Description	Alice wants to get groceries, Stacy doesn't feel like going out but would like some stuff from the store, so Alice and Stacy decide they want to split the grocery bill. When Alice gets home with the groceries, she realizes that Stacy asked for significantly more things than she got, but still wants to split the cost 50/50. This makes Alice feel cheated, and neither wants to do the math to see exactly how much is owed.
Solution	Alice can instead type in a percentage/fraction, and the bill will automatically be updated. After some negotiations, Stacy agrees to pay 65% of the bill, which Alice feels is much fairer for both of them. Everyone walks away feeling satisfied.
Diagram	

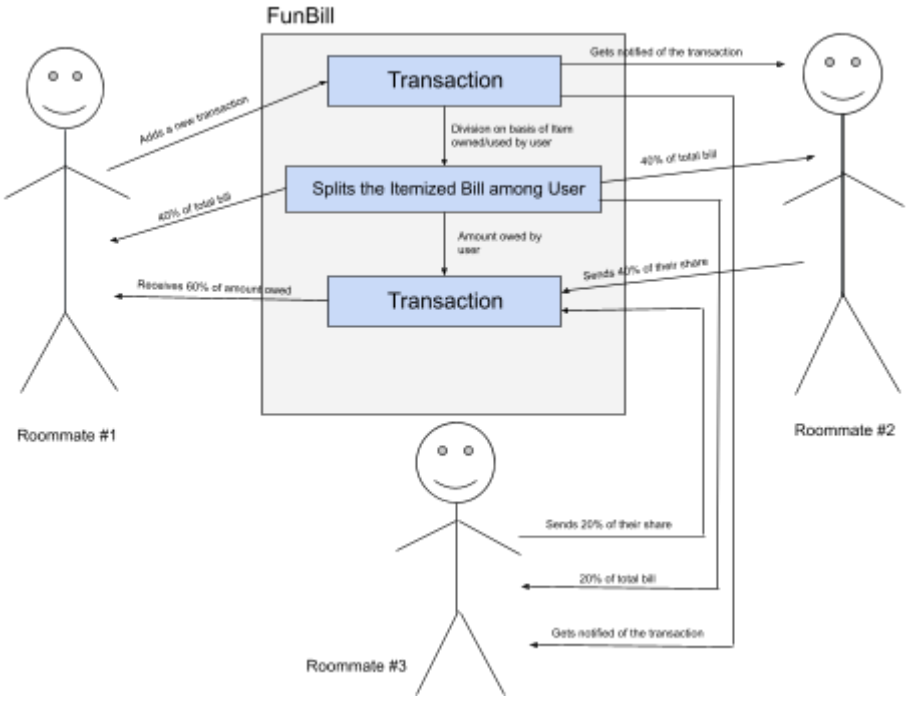
Back to Table of Contents

3. Process Payment

Use Case	Process Payment
Actors	Dave - Friend #1 (User), Mary - Friend #2 (User), FunBill (Web App)
Description	Mary is out with a group of friends she hasn't seen in a while. Included in this group is Dave, who has been given the task of being the designated driver for tonight as he doesn't drink. Mary and her friends go bar hopping throughout San Francisco, only to be too inebriated to pay for their own drinks at the end of the night. Dave then picks up the tab for Mary and pays for her share in full with his own card. The next day Mary realizes that Dave had to pay for everything and wants to reimburse him for the trouble she has caused him. However, Mary and Dave are going to be pretty busy and don't know when they will get to see each other again in person.
Solution	This is where FunBill can help, by Dave simply charging Mary for her total tab on Funbill. She can then electronically, through the application, pay Dave what she owes him. With FunBill, Dave gets paid, Mary has her guilty conscience lifted and everyone is happy.
Diagram	

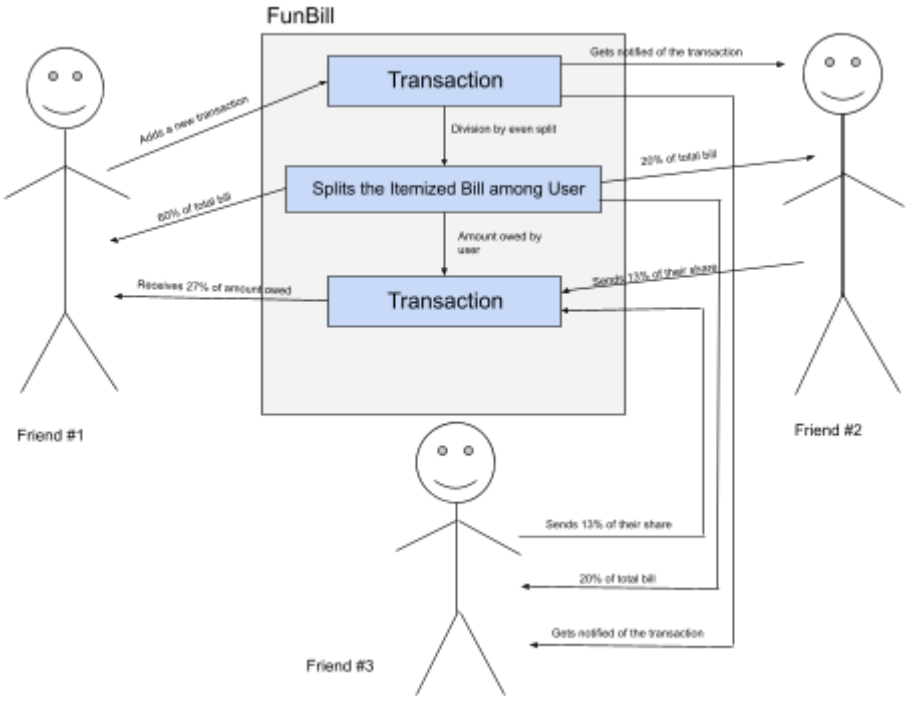
Back to Table of Contents

4. Roommates want to be fair and keep track of every month's spending

Use Case	Roommates want to be fair and keep track of every month's spending
Actors	Zlatan - Roommate #1 (User), John - Roommate #2 (User), Monica - Roommate #3 (User), FunBill (Web App)
Description	Three roommates living together buy groceries and new things every week. Zlatan decides to buy fruit and gets it for everyone in the house. But Monica doesn't like bananas and she doesn't want to pay for bananas but she is willing and wants to pay for other fruits. Calculating the portion Monica will have to pay is difficult to do manually and frustrates the three of them.
Solution	FunBill allows users to itemize bills and select which items will be split amongst the group. This alleviates the stress of math from the group of roommates. Everyone is happy as a result.
Diagram	

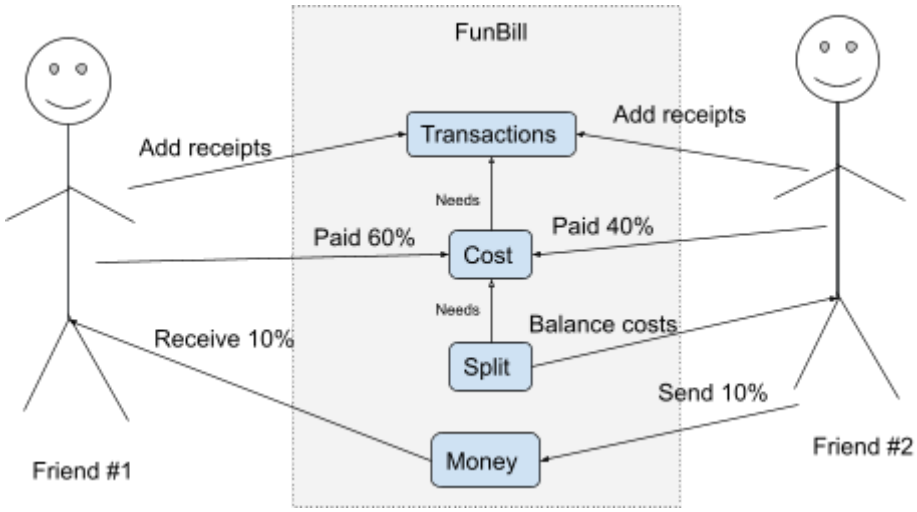
Back to Table of Contents

5. Bill split calculator on a trip with friends

Use Case	Bill split calculator on a trip with friends
Actors	Jack - Friend #1 (User), Brian - Friend #2 (User), Zack - Friend #3 (User), FunBill (Web App)
Description	Three friends: Jack, Brian and Zack are on a trip together. Over the course of the trip, the three pay for many instances of cost. With the flow of the trip being spontaneous, the three try to enjoy their time together as friends on a trip rather than stressing out on the less fun part of splitting numbers on the bills evenly. At the end of the trip, the three are confused about who owes how much.
Solution	Trying to make things fair, Zack decided that they could utilize FunBill to evenly split the costs among the three friends. Alternatively, the three could have recorded their costs through FunBill. By the end of it, FunBill will accurately calculate who owes each other how much.
Diagram	

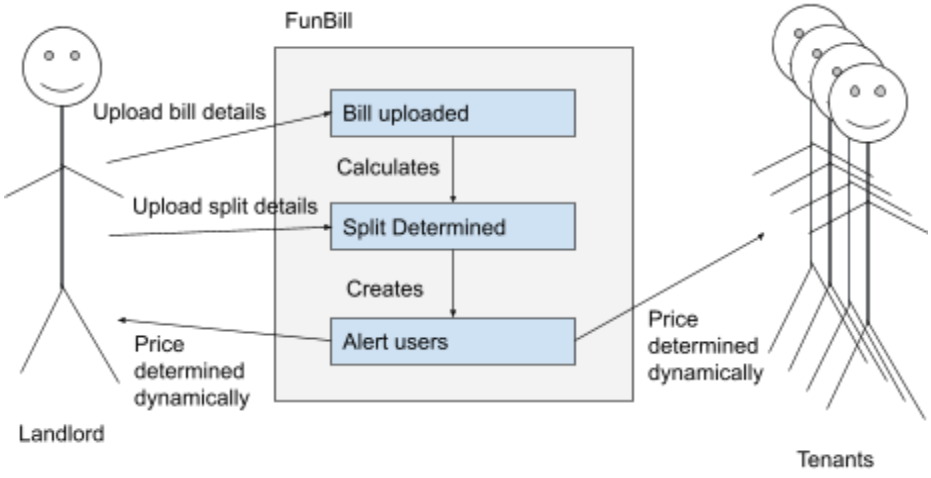
Back to Table of Contents

6. Buying items from different stores

Use Case	Buying items from different stores
Actors	Fred (User), George (User), FunBill (Web App)
Description	Fred and George are planning a surprise event for their brother Ron. They need to buy a long list of items from different stores (some stores don't have a certain item or prices are better at other stores) and want to split the bill among each other.
Solution	FunBill allows users to include multiple transactions with different proportions. Once all transactions are added, FunBill will balance out the costs to whoever paid the least.
Diagram	

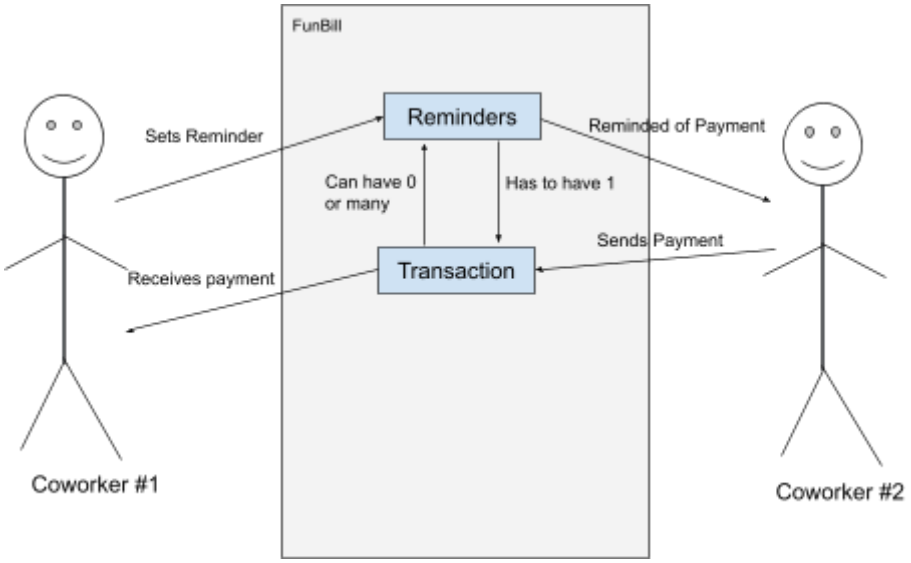
Back to Table of Contents

7. Landlord Splitting Utilities With Tenants

Use Case	Landlord splitting utilities with tenants
Actors	Jack (User), Tenants of Jack's Apartments (User), FunBill (Web App)
Description	Jack is the owner of an apartment building and wants to split half the cost of the water bill of the building with the tenants who live there while he foots the other half. He doesn't want to be seen as untrustworthy, so he needs a way to prove that the split is equal and that nobody is overcharged.
Solution	FunBill allows users to share what the split bill is and how much the total bill was before and after the split.
Diagram	

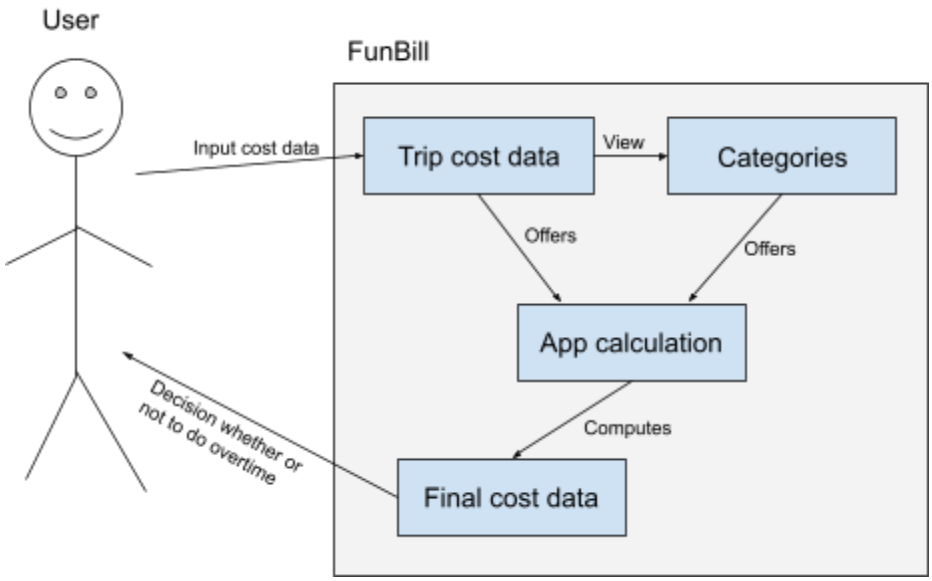
Back to Table of Contents

8. Reminders to a Forgetful Person

Use Case	Reminders to a Forgetful Person
Actors	John (User), Ben (User), FunBill (Web App)
Description	Ben is a tech employee who goes out to eat with his co-workers on lunch breaks at his company. One of his coworkers is John. One day Ben forgets his wallet at home and has no way to pay for his portion of the lunch. John picks up Ben's tab under the guise that Ben will pay him back in full later that week. However Ben can be a forgetful person, along with the fact his workload has been increased, he completely forgets that he needs to pay John back for lunch.
Solution	This is where FunBill can help, as the application has a feature that allows users to set reminders to the recipient till they follow through with the payment. So John can set a reminder to alert Ben to pay John back, reminding him and not causing a rift among coworkers. With the app, Ben was able to pay back John and everyone leaves full and happy.
Diagram	

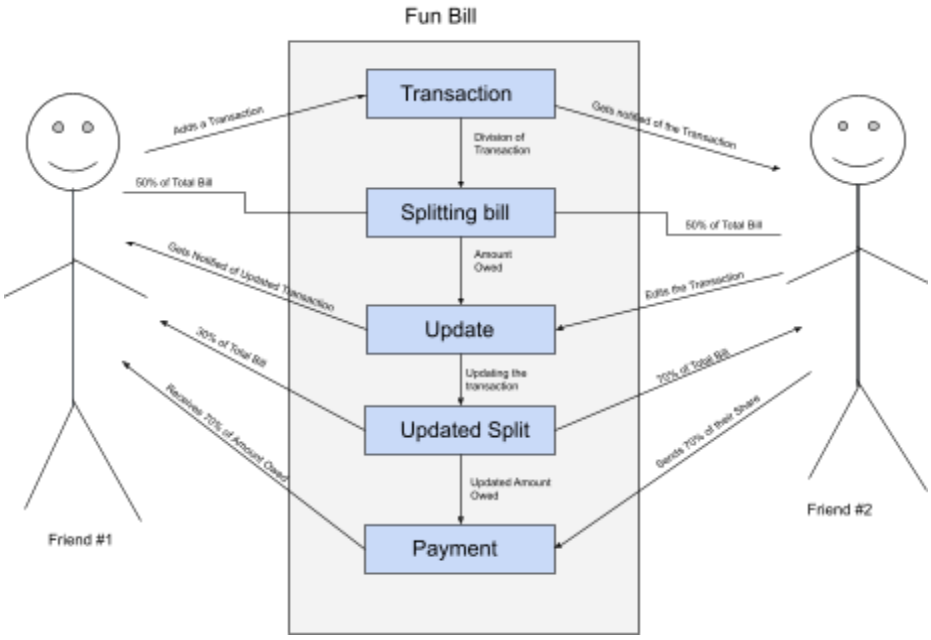
Back to Table of Contents

9. View expenses per category

Use Case	View expenses per category
Actors	Jacob (User), Funbill (Web App)
Description	Jacob is exhausted since he has recently come back from a vacation trip with friends. He wants to easily look at his expenses from the trip to decide if he needs to work overtime next week.
Solution	Funbill allows the user to see an overview of their expenses based on category and date. This way Jacob can view the expenses per category during the period of his trip.
Diagram	

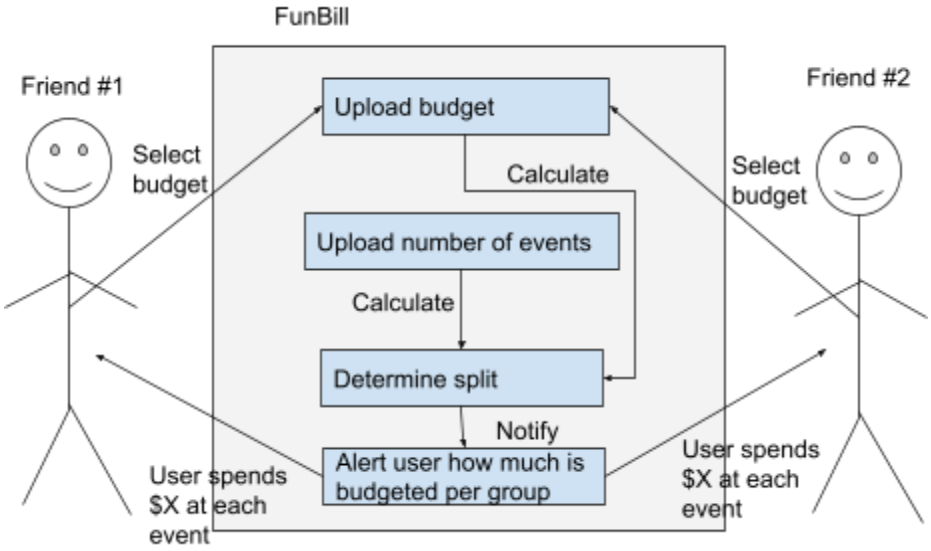
Back to Table of Contents

10. Updating a transaction by a user

Use Case	Update a transaction by a user
Actors	Steve - Friend #1 (User), Stacy - Friend #2 (User), FunBill (Web App)
Description	Steve and Stacy go out for lunch in a fancy restaurant. After lunch, Steve pays the total bill for their lunch. So Steve adds the total bill of the lunch on the app splitting equally. But Stacy feels she should pay more as she wanted to pay extra for what she had during lunch and did not want Steve to pay for her part of the bill. This makes Stacy feel bad as Steve had to pay extra for something he did not have and had to pay Stacy's part as she ordered something extra during lunch.
Solution	FunBill allows the user to update, edit or even delete the transaction and split the bill equally or unequally based on their needs. This makes everyone happy and feels satisfied knowing they only pay for what they had or are owed justifiably.
Diagram	

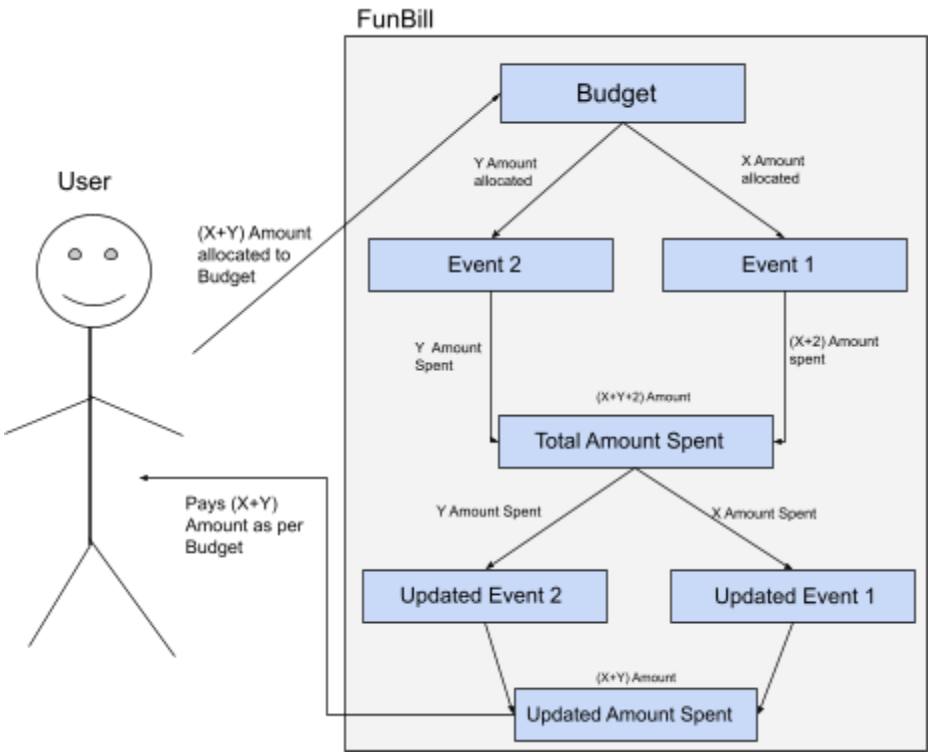
Back to Table of Contents

11. Preemptive split

Use Case	Decide the split of a purchase that hasn't happened yet.
Actors	John (User), Sara (User), FunBill (Web App)
Description	John and Sara have been stressing about a final they just took together and want to celebrate passing it. They have a budget of $100 each but want to go see a movie and go bowling afterward. They are worried that if they buy too many snacks at the theaters they won't be able to go bowling.
Solution	FunBill allows users to preemptively split a bill before it has happened in real life.
Diagram	

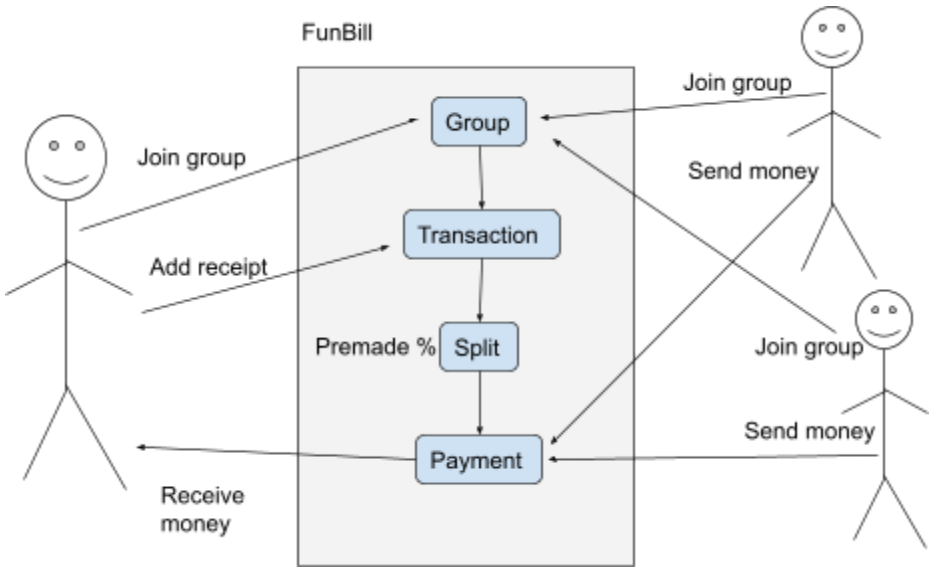
Back to Table of Contents

12. Budgeting

Use Case	Creating a budget for a trip
Actors	Yuzi (User), FunBill (Web App)
Description	Yuzi and a couple of his friends decide to go on a trip. During the trip, it comes to Yuzi's knowledge that they are going to stay in a 5-star Hotel and the budget for that is too high for what Yuzi has to offer. Yuzi is too shy to tell anyone and feels uncomfortable spending money for the rest of the trip.
Solution	The FunBill web application allows its user to keep track of their spending and allow them to create a budget to limit their spending on a particular item. Yuzi decides to cut some ends and meet everyone to stay in a 5-star hotel by creating a budget on how much he will spend on what item during the trip.
Diagram	

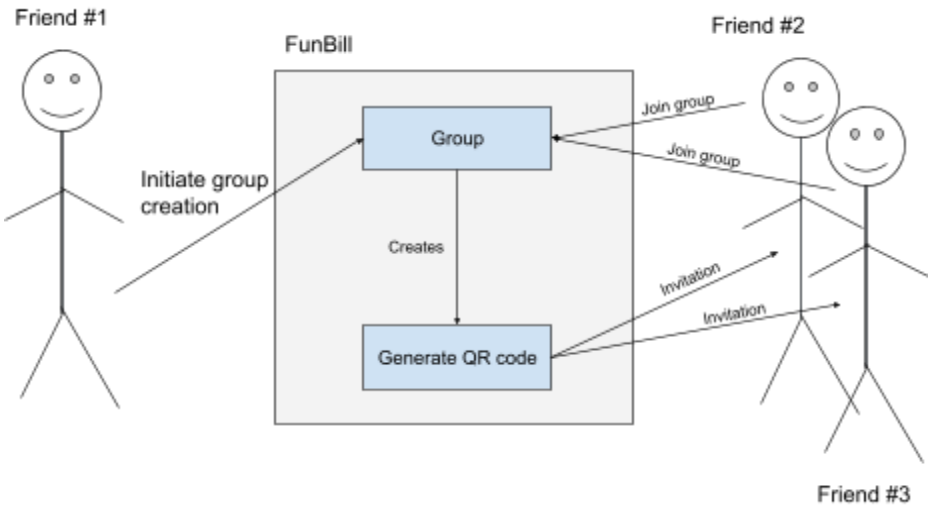
Back to Table of Contents

13. Premade Groups

Use Case	Premade Groups
Actors	Nic (User), Ray (User), Emily (User), FunBill (Web App)
Description	Nic, Ray, and Emily often go out to eat together. They're all eating out for dinner and end up the night feeling super full and satisfied. They usually split the bill evenly amongst themselves but are tired of having to keep calculating their split.
Solution	FunBill can allow users to create a group with certain people and have default splits for bills i.e. divided evenly.
Diagram	

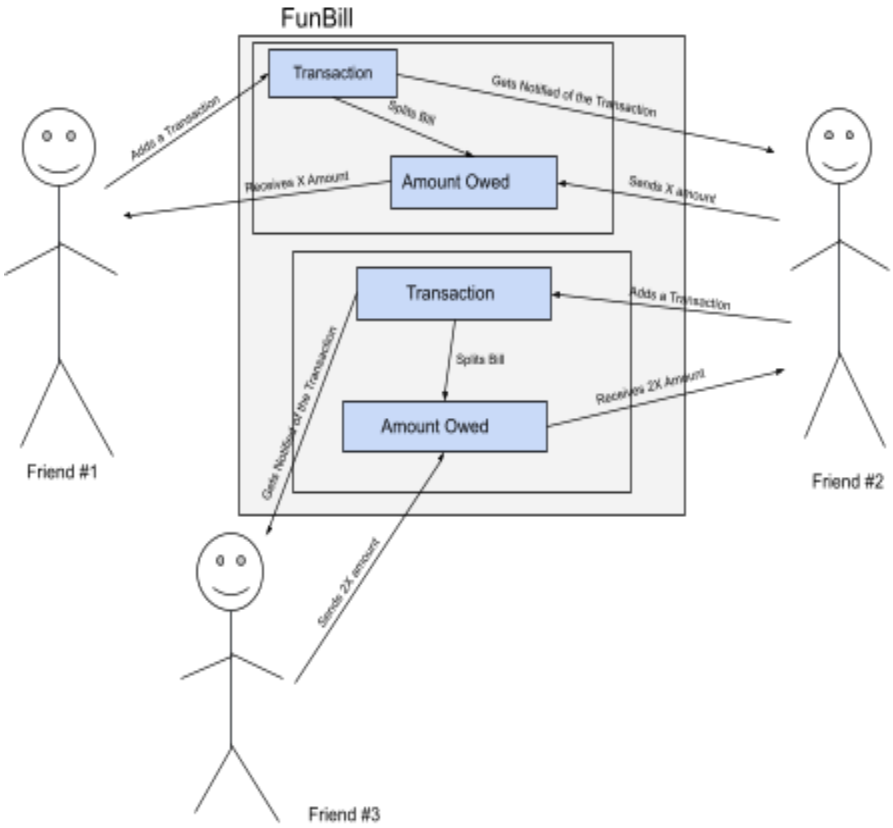
Back to Table of Contents

14. Group link with QR code/link

Use Case	Group link with QR code/link
Actors	Ryan (User), Brock (User), Jamie (User), FunBill (Web App)
Description	Ryan, Brock and Jamie were out on their weekly dinner and bowling night. Ryan notices Brock and Jamie are tipsy and offered to use his credit card during the night out. Knowing his friends for years, he expects his two friends to split to bill with him later on. It's the next day and Ryan wants to connect with his friends and calculate the bill split.
Solution	FunBill allows a smooth solution where Ryan can create a virtual group that others can join via group link or QR code. This way, Brock and Jamie are able to join Ryan's group and split the bill evenly amongst themselves. Because FunBill lets users choose an existing group and agree on a split percentage, the three friends no longer need to worry and put effort into splitting the bill. The three friends end up enjoying their weekly hangouts without a hitch.
Diagram	

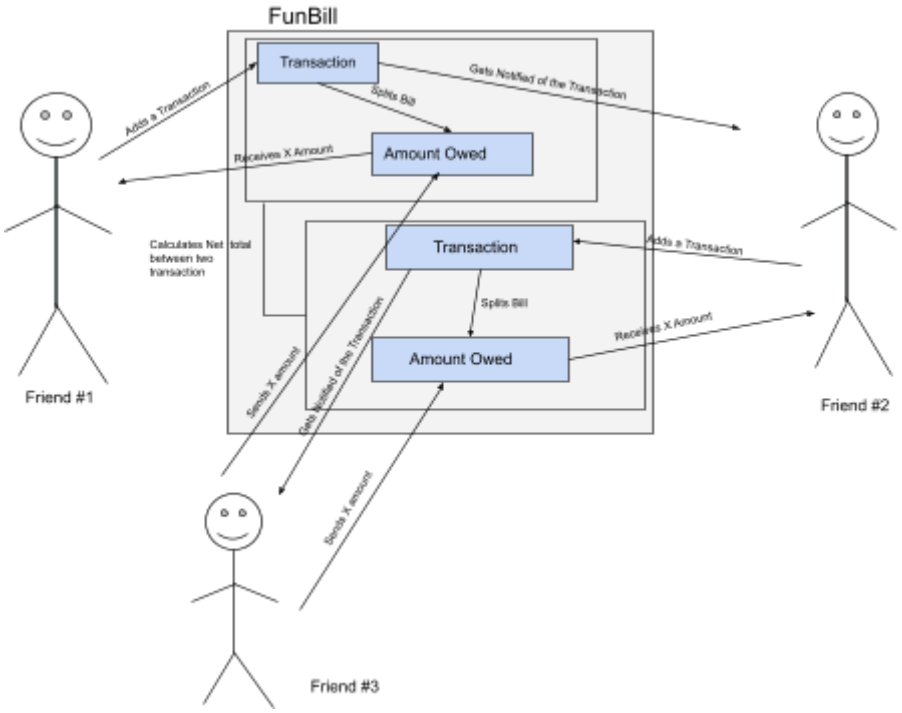
Back to Table of Contents

15. Simplify Group

Use Case	Allowing automatically combining debts in this group to get rid of unnecessary extra payments.
Actors	Tarik - Friend #1 (User), Megan - Friend #2 (User), Stew - Friend #3 (User), FunBill (Web App)
Description	A group of three friends Tarik, Megan, and Stew go for lunch. During lunch, Tarik pays for Megan's part which is (X) amount, and updates it on FunBill. After some days they go out again to party and there Megan pays Stew's part of the amount which is (2X) the amount. So Megan owes Tarik (X) amount and Stew owes Megan (2X) amount. Megan pays Tarik her (X) amount owed and then receives her (2X) amount owed by Stew. This creates too many transactions between each friend and also creates multiple transactions for sending and receiving money.
Solution	FunBill simplifies these transactions and automatically combines debts between groups of friends and gets rid of unnecessary transactions of collecting and receiving money.
Diagram	(1) General Case 

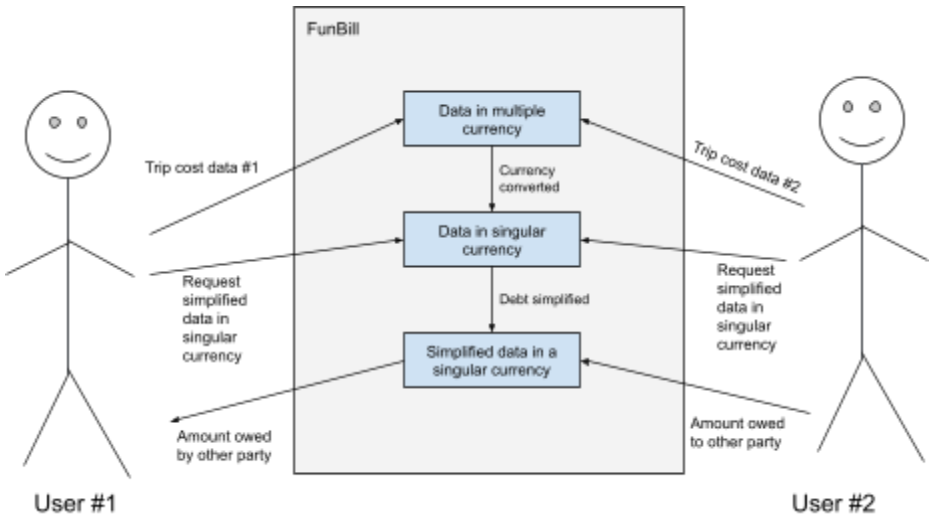
Back to Table of Contents

(2) Simplified Case



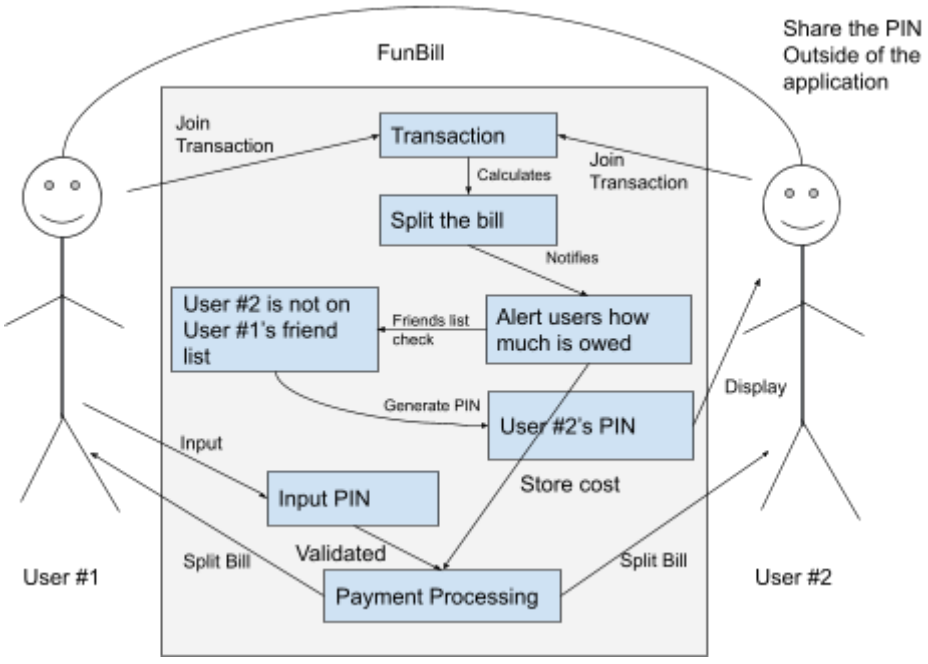
Back to Table of Contents

16. Multiple currency usages

Use Case	Traveling and using multiple currencies
Actors	James (User #1), Victoria (User #2)
Description	Victoria and her boyfriend James have been living together in the US for a couple of years now. Because both Victoria and James happened to get promotions around the same time, they wanted to celebrate by going on a huge tour of Asia. They've already visited several countries including China, Japan, and Korea. During the trip, the couple used three different currencies: CNY, KRW, and JPY. They have also been recording the costs they've spent in each country in their respective currencies. After arriving back safely in the US, the couple now wants an effective solution to figure out how much they owe each other in USD.
Solution	FunBill offers users to choose their preferred currency. Once the preference is selected, all other transactions will automatically convert to that currency. In this case, both James and Victoria want their final currency to be USD. All the transactions will convert CNY, KRW, and JPY to USD. Since the couple has many different transactions, they can choose to have FunBill apply the Overall Simplified Debt feature. This will simplify all the debts to one number. James and Victoria have had a wild adventure, and are ready to back to making that bank.
Diagram	

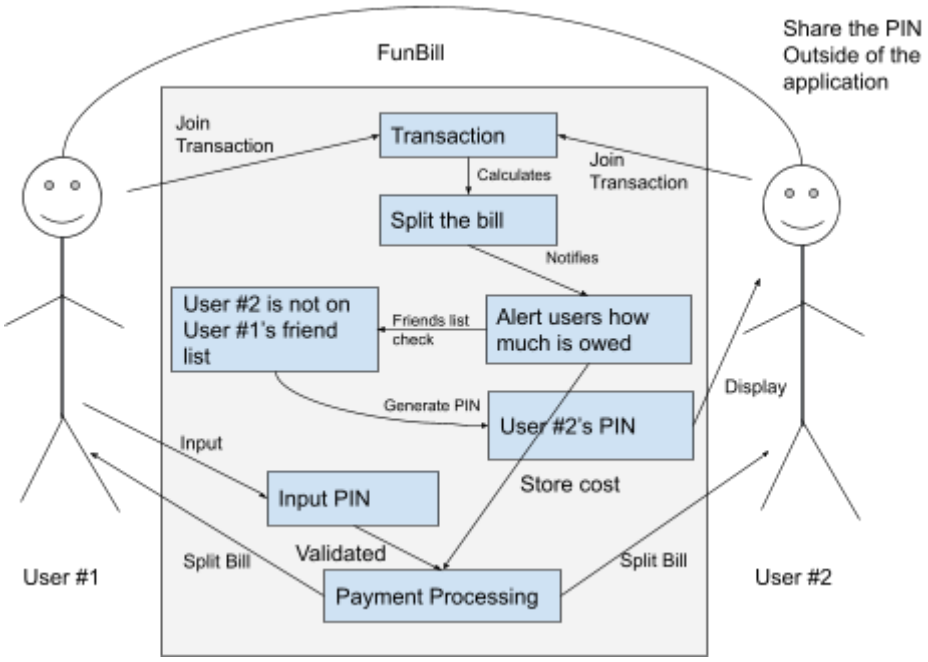
Back to Table of Contents

17. Transaction Identity Verification

Use Case	Verifying one user is who they claim to be.
Actors	Marley, Charlotte
Description	Marley and Charlotte go and get lunch together and want to split the bill. When they use FunBill, Charlotte tries to search for Marley's FunBill account and gets phished because Marley's account handle, @marley looks very similar to the account @marley (MARLEY vs MARIEY). This results in Charlotte sending money to the wrong person.
Solution	To prevent this situation, FunBill will verify users with a pseudo-two-factor authentication. By giving a temporary PIN to Marley that Charlotte can use to guarantee she is paying the right account, these issues can be avoided. Additionally, if the users are on each other's friends list the authentication can be bypassed.
Diagram	

Back to Table of Contents

18. Monthly Group Billing

Use Case	Remind the person who pays in a group about recurring expenses
Actors	Harry (User), William (User), Ash (User), FunBill (Web App)
Description	Harry, Will, and Ash are roommates and use FunBill to split their utilities each month. They all like the app but are annoyed that they have to scramble to pay rent each month since Harry has a very bad memory.
Solution	FunBill offers a way for members in a group to set up an automatic reminder for recurring payments. Now that Harry no longer stresses out William and Ash, they can all focus on who's doing the dishes this week.
Diagram	

Back to Table of Contents

Section III: Main Data Items and Entities

1. **General User:** a user that can use the web application without creating an account
2. **Registered User:** a user that has created an account within the web application
3. **Groups**: a collection of users that will make up a group
4. **FunBill Transaction:** a collection of Bills uploaded to FunBill with a total cost that is dynamically split between users or a group.
5. **FunBill Transaction Storage**: a transaction that has yet to be completed will be stored within the transaction storage.
6. **FunBill Transaction Log:** a log of all completed transactions
7. **FunBill Transaction Receipt**: a copy of the transaction online.
8. **Currency Type:** the type of currency being used in a transaction—must be a member of **Currencies Supported**
9. **Currencies Supported:** the total set of currencies supported by FunBill
10. **Currency Conversion:** a conversion between two different currency types
11. **Transaction Payment**: an exchange of currency between two or more users in order to settle a transaction, typically of the amount either dictated by the transaction or the simplified debt algorithm.
12. **Transaction Payment Type**: each payment will have a payment type
13. **Transaction Payment Request:** a request sent from one user to another to pay an amount specified.
14. **Transaction Payment Request Description:** a payment request will have a pay request description to describe the reason for the payment request.
15. **Simplified Debt:** a currency value that is derived from a series of transactions between a group of two or more users.

Back to Table of Contents

16. **Reminders**: an alert to remind the receiving user that the transaction has been given and is due soon
17. **Search**: users can use a search function to look up other users and/or past transactions
18. **Categories:** an attribute or tag given by the user to organize transactions
19. **Home Feed:** a social feature that can post/display posts of registered users who use the application. It consists of registered users' friends' posts and transactions
20. **Scan Receipt**: scan receipt retains the data that was scanned from the physical copy of a receipt.
21. **Birthday List:** a list containing all the user's friends, who have also toggled birthday lists on.
22. **Chat Box**: a social feature that allows registered users to communicate one on one with other registered users.
23. **PIN:** a verification method to allow users certainty that they are paying the right person.
24. **Admin:** a registered user with additional privileges to help keep the web application safe.
25. **Bill:** an external transaction uploaded to a FunBill Transaction to be used in the application.

Back to Table of Contents

Section IV: Functional Requirements

1. General Users

1.1. A general user shall be allowed to use the FunBill web application to participate in transactions

1.2. A general user shall not be able to change their username until they register for an account.

1.3. A general user shall have the option to sign up for an account and become a registered user.

2. Registered Users

2.1. A registered user shall be able to create transactions with other users.

2.2. A registered user shall be able to create groups and group other users together

2.3. A registered user shall be able to set a username.

2.4. A registered user shall be able to change their password at any time.

2.5. A registered user shall be able to change their profile picture.

2.6. A registered user shall be able to invite others to a transaction.

3. Groups

3.1. A group shall be comprised of one or more members

3.2. A registered user shall be able to create a group with other registered users

3.3. A registered user shall be able to add multiple transactions with multiple users

3.4. A registered user shall be able to edit/delete a given transaction they are apart of

4. FunBill Transaction

4.1. A registered user shall be able to upload a transaction

4.2. A registered user shall be able to edit/delete an existing transaction until the transaction is complete/deleted.

4.3. A registered user may be able to delete a transaction they requested

4.4. A transaction shall consist of at least one payment and at least two users, and a proportion that determines who owes how much.

4.5. A registered user shall be able to change the transaction's total cost and proportions until the payment is made.

4.6. A payment request may create a transaction

Back to Table of Contents

4.7. A registered user shall be able to invite other users to join a transaction.

4.8. Transactions shall be categorized by the user, but it is not mandatory.

4.9. Registered users shall be given the option to charge another user or mark a payment as fulfilled to allow cash transactions.

5. FunBill Transaction Storage

5.1. Registered users shall be able to see their transaction storage, to see their current debt or money owed.

6. FunBill Transaction Log

6.1. Transaction logs shall be intractable via category selection or through filtering

7. FunBill Transaction Receipt

7.1. A transaction is settled after all registered users have paid their split, and general users are marked as paid

7.2. Registered users can copy the transaction receipt and send them to other users

8. Currency Type

8.1. Registered users shall be able to choose their preferred currency type

8.2. A Currency type should be defined to be USD, Pound, Yen, Euro, other

9. Currency Supported

9.1. Registered users shall be given the option to choose a currency type among a currency selection

9.2. Payment currency shall consist of 10-20 international currencies.

10. Currency Conversion

10.1. Registered users shall be able to convert the current transaction set currency type into a different currency type

10.2. If a currency preference is requested by a registered user in a group, the transaction creator shall be able to decide to accept/decline the request

10.3. Registered users shall be able to request a simplified debt after having derived from multiple costs that were present in multiple currencies

Back to Table of Contents

11. Payment

11.1. Payments shall be made using a single currency.

11.2. Payments will be processed immediately

12. Payment Type

12.1. Payment type can include cash

12.2. Payment types can consist of Visa, Mastercard, or American Express cards

12.3. Payment types can include third party options like PayPal, CashApp.

13. Payment Request

13.1. A payment request shall be used to create transaction

13.2. A registered user shall be able to send a payment request to another registered user

13.3. A pay request must have a value over 0.00 dollars

14. Transaction Payment Request Description

14.1. Registered users shall be able to decide if they want to have a transaction description describing the reason for the payment request

15. Simplified Debt

15.1. A registered user should be able to generate a simplified debt number

15.2. A registered user should be able to include any transaction within the collection.

15.3. A registered user should be able to exclude any transaction within the collection.

15.4. A registered user should be able to attach proof accompanying transactions within the group's simplified debt.

16. Reminders

16.1. A registered user may add a reminder to a transaction for the receiving registered user.

16.2. A reminder will alert the receiving registered user at the time specified by the sending registered user.

Back to Table of Contents

17. Search

17.1. A registered user shall be able to search for other registered users to add and send money to.

17.2. A registered user shall be able to search for past transactions.

17.3. A registered user shall be able to search by a date range

17.4. A registered user shall be able to search based on a specific category

17.5. A registered user shall be able to search based on keywords

18. Categories

18.1. Registered users shall be able to give each transaction a category

18.2. Registered users shall be given category options to choose from

18.3. Registered users shall be able to make their own category

18.4. Search shall be able to sort/filter results by category type

18.5. A Category should be defined to be either a Event, Outing, Shopping, Gas, Food, Rent, or Other

19. Home Feed

19.1. A registered user shall be able to post on the home feed

19.2. A home feed shall be toggled on or off by registered user

19.3. A registered user shall be able to comment on posts on their home feed.

20. Scan Receipt

20.1. A registered user shall be able to upload an existing receipt.

20.2. A scanned receipt shall be used to produce a transaction

21. Birthday List

21.1. A general or registered user shall have the option to be added to a birthday list on the tab so that their expenses are covered by everyone else.

21.2. The birthday list shall be toggled on or off by the registered user

22. Chat Box

22.1. Registered users shall be able to chat with one another through the chat box.

22.2. Registered users shall be able to interact with previous chats

Back to Table of Contents

23. PIN

23.1. A PIN shall be generated by a transaction between users who are not on each others' friends list.

24. Admin

24.1. A admin shall be one account

24.2. An admin shall be able to delete posts from other users.

24.3. An admin shall have permissions to edit / delete accounts

25. Singles

25.1. A Singles shall be used to generate a pay request between two users.

25.2. A singles can have be linked with a reminder

25.3. A Singles can use scan receipt

26. Account:

26.1. A Account shall be able to create transactions with other users

26.2. A Account shall be able to create groups and group other users together

26.3. A Account shall be able to set a username

26.4. A Account shall be able to change their password at any time

26.5. A Account shall be able to invite others to a transaction

26.6. A Account shall be able to edit/delete a given transaction they are apart of

26.7. A account shall be able to change/edit their profile picture

26.8. A account shall be able to change/edit their password

26.9. A account shall be able to change/edit their username

26.10. A account shall be able to update their emails

Back to Table of Contents

Section V: Non-Functional Requirements

1. **Memory**
   - 1.1. The web application shall have at least 1 GB of RAM

2. **Storage**
   - 2.1. The web application shall have 20 GB of SSD storage

3. **Scalability**
   - 3.1. The web application shall support up to 500 concurrent users.

4. **Database**
   - 4.1. The database shall use MySQL
   - 4.2. The database shall be stored within Amazon's AWS online services.
   - 4.3. The database shall define the relations between entities given.
   - 4.4. The type of database used shall be relational.

5. **Security**
   - 5.1. User information shall be encrypted.
   - 5.2. User passwords must have at least 8 characters and contain at least one of the following: an uppercase letter, lowercase letter, number, and/or special character.

6. **Backup**
   - 6.1. Duplicating a set of data records for the event of backing up if and when the original data is lost.
   - 6.2. Data will be backed up every day through AWS Backup
   - 6.3. Data will be backed up using AWS Backup at UTC-08:00.

7. **Response Time**
   - 7.1. The system will not take longer than 30 seconds to process requests related to transactions. In the event the system must take longer, the system will state it is correct and add the request to a processing queue.

8. **Data Integrity**

Back to Table of Contents

8.1. Maintaining and assuring the data accuracy and consistency over a given period.

8.2. Data is guaranteed to be accurate and consistent until December 15, 2022.

9. Maintainability

9.1. Ease of maintaining a program which may include preventing unexpected faults, correcting or fixing bugs, meeting future needs, etc.

10. Coding style

10.1. The Team shall follow Google's style guide for Javascript

11. Browser version

11.1. The web application shall work on the following browser versions: Chrome, Firefox, Microsoft Edge, and Safari

12. Integration

12.1. The system must pass all tests before being deployed into production.

13. Testing

13.1. All changes will be tested thoroughly before pushing to the main branch.

14. GitHub

14.1. All commits shall be pushed to their respective branches when the thought process of the developer has been completed.

14.2. The main/master branch shall never compile with errors.

14.3. The backend branch shall have the most up-to-date version of the backend, with bugs and errors. The frontend of the backend shall be the most current bug-free version.

14.4. The frontend branch shall have the most up-to-date version of the frontend, with bugs and errors. The backend of the frontend shall be the most current bug-free version.

14.5. Pull requests shall be handled by the Github Master.

14.6. A live version of the web application shall be deployed from the main branch.

15. Policy

15.1. The site shall have terms and conditions document

Back to Table of Contents

15.2. The site shall have a privacy policy.

15.3. Users are not allowed to register until they agree with the terms and conditions and privacy policy of the site

16. Distribution

16.1. The web application shall be available for free

16.2. The web application shall be to anyone who has access to Chrome, Firefox, Microsoft Edge, or Safari.

17. Accessibility

17.1. Emphasis on the design of the application so that users can access it such as direct access including web accessibility.

17.2. The web app shall have various color themes that support color blindness

18. General Users

18.1. General users shall be provided with a temporary unique ID key generated by the database, the default will be of the format @FIRSTNAME.LASTNAME<NUMBERS>

19. Account Security

19.1. An account shall be linked to a registered user.

19.2. An account shall have a unique username.

19.3. An account shall have a password.

20. Registered Users

20.1. Registered users must have a unique ID key

20.2. Only one non-existing email shall be permitted to create a registered user

21. Transaction storage

21.1. Transactions shall be stored using browser cookies and linked to all involved Registered Users' accounts.

22. Transaction composition

22.1. Transactions shall be made of *n* purchases where *n* has no upper cap.

Back to Table of Contents

22.2. Transactions may include a PIN-generated PIN. PINs will only be generated at payment processing-time, and if a transaction is handled by FunBill.

23. Transaction Description

23.1. Transactions may have a description

23.2. Transactions must have a currency set

23.3. Transactions must have a monetary numeric value set

23.4. Transactions will be assigned a date and timestamp based on when it was generated.

23.5. Transactions must have a unique numerical identifier

23.6. Transactions must have at least one recipient

23.7. The user cannot be both the sender and recipient

24. Transaction Receipt

24.1. Each transaction completion shall produce a transaction receipt for the history of the transaction

24.2. The transaction receipt can be found via its unique numerical identifier

25. Transaction History

25.1. Each registered user shall have a transaction history

25.2. Transaction history shall be viewable to only the user that owns it

26. Pay Request Description

26.1. A pay request must have a pay request description in the form of alphanumeric characters.

26.2. A pay request description shall be limited to 180 characters.

27. Scan Receipt

27.1. Scanned receipts will be stored inside the database

27.2. Scanned receipts shall be only used for transaction calculations

28. User Reminders

28.1. A reminder shall only have 1 transaction and a transaction shall have 1 or 0 reminders.

Back to Table of Contents

28.2. A user has the right to make a reminder to the other party when it comes to pending payments after a set period of time

29. System Reminders

29.1. A Pay Request that has not been met after a specified reminder deadline will be resent to alert the receiving user.

29.2. A Pay Request Reminder has a cooldown of 24 hours.

30. Home Feed

30.1. The home feed shall consist of user uploads, and transactions including the user's friends

31. Filtering

31.1. Transaction filtering can be done by the following categories: transaction category, users involved, date, total cost, and cost per user

31.2. User filtering can be done by the following categories: user's name, username, and phone number

31.3. Multiple filters shall be allowed to be layered on top of each other.

32. Friends List

32.1. A registered user shall be required to add friends to have social features/interactions.

32.2. A registered user who is friends with another user may send money bypassing verification.

32.3. Registered users that are not on each other's friends list must enter a randomly generated six-digit PIN before any payments can be made. Both users will generate PINs for their clients, but only one user will be required to enter the other's PIN to verify identity.

33. chatData

33.1. A chatBox shall store its chat conversation within chatData

33.2. chatData shall store data by using browser cookies

33.3. chatData shall be linked to all involved Registered Users' accounts

Back to Table of Contents

Section VI: Competitive Analysis

Competitor Summary

1. Splitwise

Splitwise is an online application available for free on mobile and desktop, with certain features only available through a paid subscription. It is an application that keeps track of users' spending based on what they input, which is kept in logs, and shows/tracks bills that have been split among two individual users or groups of users. The targeted audience of the very application is the coming generation which can be categorized but not limited to Students, Roommates, Friends, Partners, Colleagues, and so on. The application promises an easier way to keep track of every bill and every spending they do on a day-to-day basis and split it with other registered users. Every user has the right to edit and add multiple transactions based on their spending and involvement with other users. At the end of the month, individual users are reminded of their spending via email. Included in the email is the amount owed to other users and the amount other users owe to them. Splitwise also has a payment option within the application itself, which helps users to settle within the application. The app doesn't set a time limit for a transaction to be completed, giving users all the time they need based on their internal agreement with the other users involved. The application is active through social media platforms like Twitter, and Instagram, and even has its own blog posts.

2. Venmo

Venmo is an online platform mainly used for transferring money between users. It is available for free on both mobile and desktop. The two main features available for users are pay and request. Users are able to link either their US bank account or debit card which the application, which Venmo can either derive currency from or transfer currency into. The main audience for Venmo consists of United States residents, as Venmo only accepts transactions in USD. Venmo is also known for having social features, such as the ability to add each other to a friends list based on usernames or phone numbers. The users are also able to add each other by scanning QR codes. The friend feature allows users to have easy access to other users who they commonly interact with, allowing for quick requests or payment transactions. Furthermore, the social feature allows users to share their transactions with those on their friend lists. Users can interact with the social feature by liking or commenting on interactions. If users don't like the social features, they can choose to opt out.

Back to Table of Contents

3. KittySplit

Kittysplit is a Berlin-based company with a free online web application that can be accessed through mobile and desktop browsers and allows anyone to create a transaction and split bills. Sharing the bill is quick and easy since it only involves sending a link to other people. Anyone with access to the bill is able to add how much they spent, and after the proportions are calculated everyone will see how much they owe to whom. Kittysplit is primarily marketed to friends doing things together and sharing costs amongst them. The website's first example is a ski trip with three friends, so it can be inferred that their target demographic would be people in their mid-20s to 30s who regularly go and do things with friends. The app is not very active on social media, nor do they promise anything extreme—it's an online bill calculator that allows users to share expenses.

4. Settle Up

Settle Up is a free transaction calculating application that also keeps track of the user's expenses. The app is available for free on mobile and desktop, but with ads, which can be remedied by buying a premium account. It works best with groups or event organizers, people who generally go out a lot in groups or have to manage others. This demographic uses the app to accurately calculate the total expenses a group has made and then share the individual cost with the rest of the group, whether the user decides to divide by an amount or by shares a person has taken. This way no one is left confused about the charge a person hands them and can see that everyone is paying the same amount, or a different amount depending on how the transaction creator has set it up. To start, you simply create a group, add the expense, calculate how it would be split (by shares or amount), and then send the individual total to each group member. Overall, Settle Up promises a cleaner, simpler and easier way to calculate individual costs in a group. This application works on all platforms and not every single person in the group needs to have the application to pay. Settle Up also has its own social media, but can be viewed only through its main website. On the application itself, you can't find any references to their social media.

5. Tab

Tab is an app solely available on Apple or Google Play store. It allows its users to create a tab by scanning a receipt or through manual input. People then join that tab and claim the items they bought. The app will then calculate each person's costs, including tips and taxes. People can pay with cash, or through the built-in Venmo integration (which as of 9/18/22 is bugged). Registered users are able to see the history of their tab, which can be shared, edited, or deleted. There is a unique birthday feature where people can be added to a list, and their costs will be evenly split among everyone else, not on that list. A feature Tab does not have is a group feature (a feature where a premade group can be made), but allows multiple users to be added to a bill. The app has social media on various platforms but has not been active in years.

Back to Table of Contents

Table 1 - Important Features

Feature	Splitwise	Venmo	KittySplit	Settle Up	Tab
Strengths	-Multi-platform Application -Easy to Use and Good User Experience -Can create multiple groups with multiple users -Has its own Splitwise Pay which allows users to send and receive money within the application -Also allows user with other payment options like cash payment and or via PayPal -Has option to upload receipts for a particular transaction -Not limiting to only one type of currency unit -Allows user to chat within group or even in a single transaction	-Easy to navigate -Multi platform: mobile and web -Allows users to add each other easily through a username -Has a QR code for social feature of adding each other onto friendlist -Has social feature of sharing transaction to your friends -Simple feature allowing the display of history -Allows a business profile, enabling users to have buyer protection -Has cryptocurrency features -Has a venmo credit card, allowing for	-Easy to use (One click to create a Kitty) -Works on all modern browsers -Email or cookies to store transactions -Website does not handle payment (Safety, can use cash/preferred methods) -Currency changes with rates based on day of purchase -No need for membership to create a transaction	-Works great in groups -Easy to use, there are three steps to sending a transaction amount -Works on all platforms -Group members don't need to have the app to receive their individual total -Can create a account with Google, Facebook, Apple or Email	-Reads items from receipts through camera scan or picture upload; very accurate -Join bill from own phone without account -Easy to set which items belong to who -Saved bills to share/edit/delete -Email with a secure link for login

Back to Table of Contents

Feature	Splitwise	Venmo	KittySplit	Settle Up	Tab
	-Lists each Transaction by category based on the type of transactions -Can export a sharable transaction with other user into single pdf -Has different fairness calculators such as rent-splitting calculators among roommates -Has option for recurring expenses	usage with Venmo credit -Easy sign-in feature linked with Facebook			
Weaknesses	-Modifying or editing a transaction is too easy in terms of security (changing or removing oneself from a transaction creates security threats) -Does not allow use of multiple current in a single instance -Allows user to clear, delete or	-Does not have bill split calculation feature -“Bill splitting” is done by direct transfer of money between users -Has to load up currency onto the account for usage for currency transfer -Does not necessarily keep track of	-No payment processing (payment must be done in a different method) -Inability to change the base currency if multiple currencies are used in the Kitty -All members must be emailed a link for a Kitty each time (No grouping of members for	-Cannot calculate payments between two people (needs a group of three at least) -Cannot make direct electronic payments through the app, it's just a calculator -No discussion or chat box in the app for in-app communicatio n about	-Weak social media presence; inactive for over 5 years -Requires phone usage for camera access to read receipt -No other payment integration besides Venmo -No in-app communicatio n between users

Back to Table of Contents

Feature	Splitwise	Venmo	KittySplit	Settle Up	Tab
	edit any transaction without consent of the other user/users involved -Does not have control over the third-party app payment and glitches mid-payment if the user is not verified or has some issue with the device -Doesn't keep track to bills or use simplify feature with friends outside a group but within the friend's list	expenses with evidence such as receipts -Only available within the United States and USD currency	repeated transactions) -Lack of user accounts.	charges. -Has bugs where sometimes the premium account features are not usable by premium account users.	
Pricing	Free with Ads Pro Account : $29.99/Year OR $2.99/Month	Free. Business: fee of 1.9% + $0.10	Free, Upgrade of $1.50 per user for extra features	Free with Ads Premium Account: 0.99$ a month	Free
Social Media	Twitter, Facebook, Instagram, Splitwise Blog	Facebook, Instagram, Twitter	Twitter, Facebook, no built-in social media	Facebook, Twitter, Instagram, but no build-in social media	Facebook, Twitter
Onboarding experience	Easy to use, easy sign-up process, flexible UI along with understandabl	Easy to sign up using Facebook and email.	Incredibly easy, the start button is in the middle of the home screen and takes the	Easy, the application shows a step-by-step process to create a group	Easy signup, no tutorial

Back to Table of Contents

Feature	Splitwise	Venmo	KittySplit	Settle Up	Tab
	e features.		user to transaction details immediately.	and calculate a transaction split.	

Table 2 - Key Features

- Feature does not exist; + Feature exists; ++ Feature is superior						
Key Features	**Splitwise**	**Venmo**	**KittySplit**	**Settle Up**	**Tab**	**FunBill**
User Account and Registration[1]	++	++	-	++	++	+
Home Feed[2]	-	++	-	-	-	+
Search Bar[3]	+	+	-	-	-	++
Filtering[4]	-	-	-	-	-	++
One on one transactions[5]	+	++	+	-	+	+
Group Splitting	++	-	+	++	-	++
Update Payment requests[6]	-	-	-	-	-	++
Currency Conversion[7]	-	-	++	-	-	+
Multiple Currency Use[8]	-	-	+	+	-	++
Payment Processing[9]	+	++	-	-	+	+
Chat within Transaction[10]	+	-	+	-	-	+

[1] Do new users have to create an account? How easy is it for guest users to become registered users?
[2] A social element allowing registered users to see transactions that their friends are a part of (e.g. Venmo).
[3] Includes users and transactions based on date range and categories.
[4] Filter options on the current feed, such as search results and on the home page.
[5] Having transactions between two people.
[6] The ability to create groups and split the pay among the members of the group. Once a group has been made, it can be used again in future transactions.
[7] Users can use more than one currency in a transaction. If multiple currencies are used in a group split, the totals will convert to a single currency.
[8] If someone wants to edit, modify, or delete the transaction, everyone involved should consent to the change.
[9] Ability to handle payments within the app. Integration with different payment methods (e.g. Venmo, Zelle, CashApp).
[10] If a certain currency is desired, transactions can convert to that currency, before calculating the split.

Back to Table of Contents

- Feature does not exist; + Feature exists; ++ Feature is superior						
Key Features	**Splitwise**	**Venmo**	**KittySplit**	**Settle Up**	**Tab**	**FunBill**
Reminders[11]	+	++	-	+	-	+
Overall Debt Simplification[12]	-	-	-	-	-	++
Receipt scanner[13]	+	-	-	+	+	+
Listing by Category[14]	+	-	-	-	-	+
Birthday!![15]	-	-	-	-	+	+
User Feedback and Support[16]	+	+	+	+	+	++

[11] Users can make comments within a transaction.
[12] Debt simplification of between multiple users spanning over different groups. Example: Inside a group, Annie owes Debby $20, and inside a different group Debby owes Annie $20. This simplification feature will calculate that Annie and Debby do not need to give each other money. Both users are happy.
[13] Ability to scan receipts and gather important items like the type of item and its price.
[14] Categories can be assigned to splits, which the user can check. The categories will contain previous logs that were tagged with that category.
[15] People on the birthday list can choose to have their debts covered by their friends. Friends can choose toggle this on or off.
[16] When tickets are placed, or if a user files a problem, they will be given some sort of progress bar or checkpoint bar. This eliminates the guessing game users have to play just to get their problems solved.

Back to Table of Contents

Key Features to be Implemented

3 aspects that none of the competitors don't have, to be implemented

- Filtering
    - Filter options on the current feed, such as search results and on the home page.
    - For example, when searching within a date range, the filter can order the results based on the most recent, oldest, highest transactions, or lowest transactions.
- Overall Debt Simplification
    - If there are debts between 2+ people, FunBill will do the math to simplify all current debts even if the debts are outside the group between two individual users.
    - For example, friends A and B both owe $10 to friend C. From a previous split, friend A owes friend B $10. The Overall Debt Simplification feature would simplify the debts so that now friend A owes friend C $20, and friend B no longer has to pay friend C. For more info, see Use Case #15.
    - Another example: Inside one group, Annie owes Debby $20, and inside a different group Debby owes Annie $20. This simplification feature will calculate that Annie and Debby do not need to give each other money. Both users are happy.
- Update requests
    - If someone wants to edit, modify, or delete the transaction, everyone involved should consent to the change
    - For example, if friend A thinks friend B is asking for too much money then friend A can request to update the transaction to an amount friend A deems fairer.

1 improvement on an aspect that competitors have, to be implemented

- User Feedback and Support
    - When tickets are placed, or if a user files a problem, they will be given some sort of progress bar or checkpoint bar. This eliminates the guessing game users have to play just to get their problems solved.
    - For example, once a ticket has been placed, a checkpoint bar will be given so that the user can check the ticket’s progress. Once the problem has been addressed/solved FunBill can notify the user saying they’ve helped improve the app.

Back to Table of Contents

Section VII: Tech Stack

Server Host:	Amazon AWS (1vCPU 1GiB RAM)
Operating System:	Ubuntu v16.04 Server
Database:	MySQL v8.0.3.0
Web Server:	Node.js v16.17.0
Server-Side Language:	Javascript v1.5
Client-Side Language:	Javascript v1.5
Supported Browsers:	• Chrome • Firefox • MS Edge • Safari
Application Programming Interfaces (APIs):	• Dwolla (In-app payments) • Wise (Currency handling) • JQuery (Media handling)
Additional Technologies:	• Web Framework: Express.js • IDE: VS Code • Web Analytics: AWS Business Analytics • SSL Cert: Lets Encrypt (Cert Bot) SASS: 3.5.5

Back to Table of Contents

Section VIII: Checklist

Status	Item	Comments
DONE	Team found a time slot to meet outside of the class	Mondays @ 8:00 pm and Saturdays @ 12:00 pm on Zoom
DONE	GitHub master has been chosen	Tolby is our GitHub master
DONE	Team decided and agreed together on using the listed SW tools and deployment server	Refer to Section VII
DONE	Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	The team has agreed to freshen up/learn all the necessary frameworks.
DONE	Team lead ensured that all team members read the final M1 and agree/understand it before submission	Made sure to have two rounds dedicated to reviewing. The first round allowed for feedback/comments. The second round revised all the items, using those comments.
DONE	Github organized as discussed in class (e.g. master branch, development branch, a folder for milestone documents, etc.)	Folders were premade, all we did is maintain the folders and create files within the folders when necessary

Back to Table of Contents

Section IX: Team Contribution

Team Member:	**Denyse Luzon**
Score:	10
Use Cases:	- Formatted the section - Worked on numbers: 3, 9, 16 - Gave feedback and edited other people's use cases
Functional Requirements:	- Rearranged and organized - Wrote 10+ - Helped edit others' requirements
NonFunctional Requirements:	- Rearranged and organized - Wrote 10+ - Helped edit others' requirements
Competitive Analysis:	- Set up and organized the tables - Gave feedback on both the summaries and tables
Features:	- Suggested a couple features - Mainly guided the discussion on key features
Prototype:	- Set up the server instance - Worked on Denyse's about page

Team Member:	**Bhagdeep Sandhu**
Score:	9
Use Cases:	- Worked on numbers: 3, 8 - Gave feedback on other people's use cases
Functional Requirements:	- Formatted the section - Wrote 15+
NonFunctional Requirements:	- Formatted the section - Wrote 17+
Competitive Analysis:	- Wrote the summary and table info for Settle Up
Features:	- Suggested some features
Prototype:	- Worked on Bhagdeep's about page

Back to Table of Contents

Team Member:	**Faiyaz Chaudhury**
Score:	10
Use Cases:	- Created the diagram template - Worked on numbers: 1, 2, 7, 11, 17 - Gave feedback on other people's use cases
Functional Requirements:	- Wrote 10+ - Helped edit others' requirements
NonFunctional Requirements:	- Wrote 13+ - Helped edit others' requirements
Competitive Analysis:	- Wrote the summary and table info for KittySplit
Features:	- Suggested many features - Gave a definition to many of the features
Prototype:	- Worked on Faiyaz's about page

Team Member:	**Kaung Nay Htet**
Score:	5
Use Cases:	- Worked on numbers: 6, 14, 16
Functional Requirements:	- Wrote 3+ - Helped edit others' requirements
NonFunctional Requirements:	- Wrote 3+ - Helped edit others' requirements
Competitive Analysis:	- Wrote the summary and table info for Venmo
Features:	- Suggested some features
Prototype:	- Worked on Kaung's about page

Back to Table of Contents

Team Member:	**Poornank Purohit**
Score:	10
Use Cases:	- Created the diagram template - Worked on numbers: 2, 4, 5, 10, 12, 15 - Gave feedback on other people's use cases
Functional Requirements:	- Wrote 10+ - Helped edit others' requirements
NonFunctional Requirements:	- Wrote 10+ - Helped edit others' requirements
Competitive Analysis:	- Wrote the summary and table info for Splitwise
Features:	- Suggested many features - Advocated for many of the features
Prototype:	- Worked on Poornank's about page

Team Member:	**Tolby Lam**
Score:	8
Use Cases:	- Worked on numbers: 5, 6, 13 - Gave feedback on other people's use cases
Functional Requirements:	- Wrote 5+ - Helped edit others' requirements
NonFunctional Requirements:	- Wrote 7+ - Helped edit others' requirements
Competitive Analysis:	- Wrote the summary and table info for Tab
Features:	- Suggested a few features - Came up with some unique features
Prototype:	- Worked on Tolby's about page

Back to Table of Contents