SENG2021 DELIVERABLE 2

Team l33t-hackerz

Mentor: Christopher Joy

z5115344 Alexander Georges
z5113751 Veeraj Sharma
z5205003 Allen Wu
z5207915 Monica He
z5190299 Sheina Edeline Tengara

Table of Contents

INTRODUCTION

This second report outlines the development of the application's design such as the software architecture that will be used and the flow of interaction between users, frontend and backend parts. Further refinement has been made to the project scope, and implementation design has been furthered. The team also made progress on business-perspective analytics to ensure that the scope is valid and has a clear use case and usability in the context that it has been built and designed for.

SOFTWARE ARCHITECTURE

The figure below represents the application's software architecture which consists of the selected Web stack, external data required for backend, languages and platform chosen to start the implementation of the application. Advantages and disadvantages for each software component chosen is discussed further below the figure.
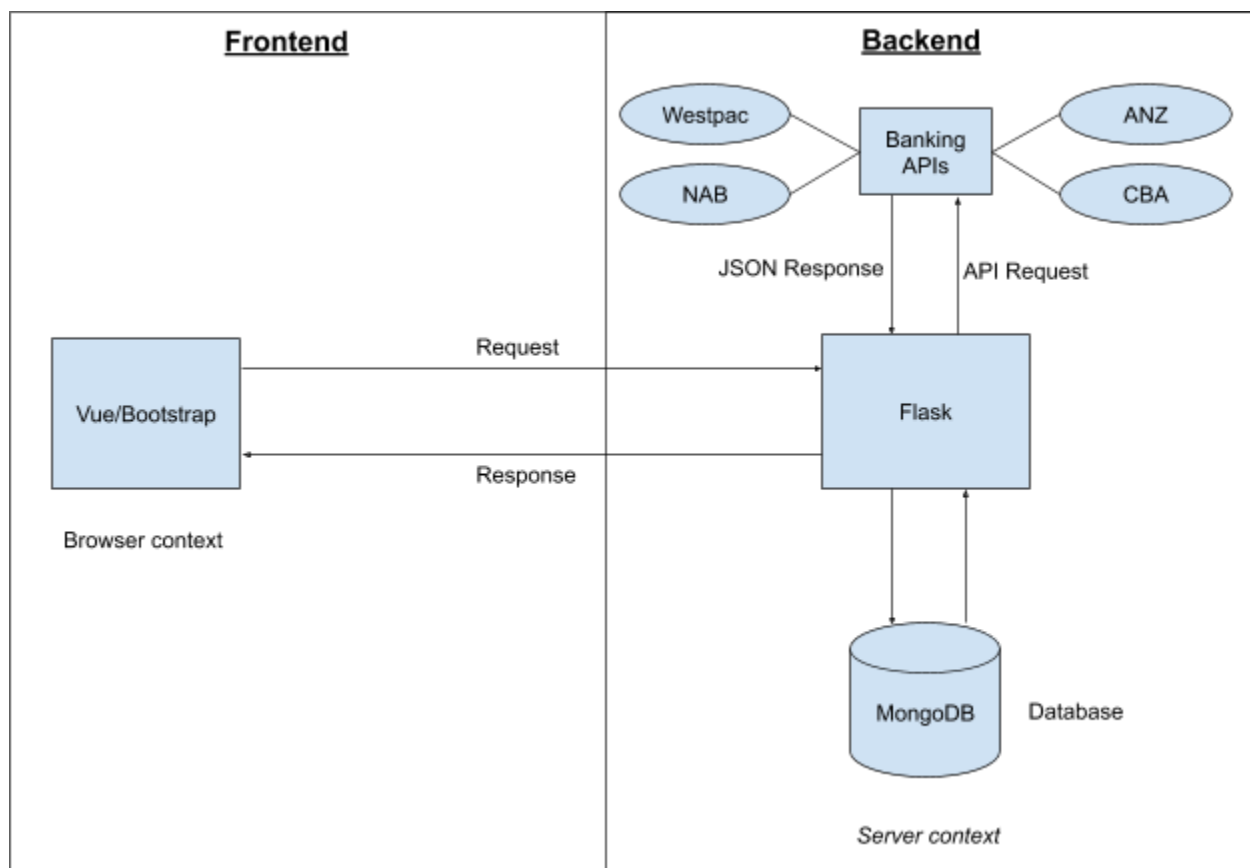
FRONTEND

Vue:

Vue is a JS framework utilizing various tools for user interface development. Our prime reasons for picking Vue for frontend development were the relatively simple learning curve, flexibility, its MVVM structure and its easy integration and compatibility with pre-existing templates. Our decision to pick Vue over other JavaScript frameworks was based on the learning curve being less steep than other frameworks like React as well as the great documentation that Vue has. The flexibility of Vue means that we can write functioning code in a variety of templates such as HTML and pure JavaScript using virtual nodes. The Model-View-ModelView pattern that Vue is used in allows us to two-way-bind data within the views meaning any changes to the model will reflect in the views and vice-versa. This ultimately makes abstraction from models safer and easier. Furthermore, Vue is easy to integrate with our pre-existing bootstrap templates meaning there is no need to do extra work on the frontend that hasn't been done already.

Bootstrap4:

Bootstrap is a CSS and HTML based frontend framework that is used for creating dynamic websites and web applications. Our reasons for picking Bootstrap4 are its easy-to-use structure, responsiveness, consistency and customizability. Bootstrap is a convenient framework to use since it has pre-built blocks of code therefore reusability is more possible and code complexity is reduced. It is responsive therefore the display can fit in any screen resolution. Since it uses pre-built code, Bootstrap4 can be worked on by different developers and still be mutually understandable owing to the consistency of its functions. However, it still offers a lot of variance and customizability as it has a large library of functions which will give us a lot of elements to work with for our frontend design. Furthermore, the documentation for Bootstrap is very comprehensive and allows us to grasp key concepts and utilities in a limited timeframe.

BACKEND

MongoDB:

MongoDB is a free NoSQL document-oriented database platform. It is compatible with many languages including Python and it stores data in JSON-like documents which eases linking of data to the backend for processing, storing, retrieving and updating information. Since data will be saved in JSON files, having JSON-like database documents would be beneficial. Moreover, the speed of accessing data is higher than a relational database due to using internal memory for storage. MongoDB is very easy to scale out and thus allows future possible expansion of the application within the banking industry. MongoDB also allows flexibility for the schema, which is ideal given the agile development cycle.

Flask:

Flask is a Python web development framework. It currently is one of the most popular web frameworks for Python and thus has a plethora of resources and documentation. Flask is a very lightweight and versatile framework which has the capability to be utilised in conjunction with MongoDB, which will be the database which we are utilising. The backend will utilise Python to handle the API calls as well as storing data in the MongoDB database and thus the most suitable framework for python would be Flask.

Open Banking API Endpoints:

The Australian Government, in conjunction with ACCC and Data61, are rolling out regulations to enforce Australian banks to comply with Open Banking standards, as the world moves towards having open industries to facilitate competition and encourage innovation and growth. Banks must comply by producing the endpoints as specified by the regulations. These are being rolled out in phases, with major banks expected to meet deadlines earlier than smaller banks and credit unions.

As of now, the major banks comply with requirements to allow access to API's to get non-customer specific product offering information. Over the next two years, banks will develop endpoints to share with other providers customer information, including their personal information, banking accounts, and transaction details.

These endpoints can be capitalized on by Fintechs, who currently only have access to this data via unsecure (and often imprecise) methods such as screen scraping.

USER STORIES

ID	User Story 1
Feature	Registering a user
Description	**As a first-time user**, I want to be able to register for an individual profile account, **So that** I can access and use the application.
Flow	**Given** that I am on the homepage, **When** I click on the 'Sign Up' button, **Then** I am redirected to a signup form where I can create an account by entering: • Full name; • Email address; • Password.
Priority	Must have

ID	User Story 2
Feature	Create a group
Description	**As a user**, I want to be able to set up a group to represent a financial living arrangement, **So that** I can utilize the application to manage joint expenditures with my housemates.
Flow	**Given** that I am on the 'Landing' dashboard, **When** I click on the 'New Group' button, **Then** I am directed to a page where I can create a group by entering: • Group name; • Members to add to the group (by username or email).
Priority	Must have

ID	User Story 3
Feature	Add and remove users to a group
Description	**As a group admin**, I want to be able to add or remove users to a group, **So that** I can share expenses with relevant people.
Flow	**Given** that I am on the 'Group Overview' dashboard, **When** I click on the 'Update Group' button, **Then** I can: • Add a group member by entering their username or email; • Remove a group member by pressing the '-' next to their name.
Priority	Should have

ID	User Story 4
Feature	Notify new members
Description	**As an added member**, I can be notified that I have been added to a group via an automated email, **So that** I can join the group and utilize the application.
Flow	**Given** I am not already a member of a group, **When** a group admin adds me to their group, **Then** I receive an email notification to inform me that I have been added.
Priority	Could have

ID	User Story 5
Feature	Updating profile details
Description	**As a user**, I want to be able to change my personal details, **So that** they can reflect my current personal information.
Flow	**Given** that I am on the 'Landing' dashboard, **When** I click on the 'Settings' button on the top right hand corner and select 'Edit Profile', **Then** I can update my personal details.
Priority	Could have

ID	User Story 6
Feature	Reviewing and updating held bank consents
Description	**As a user**, I want to be able to review and update the consents granted by me for my bank to share my account data, **So that** I can manage which of my data is being shared.
Flow	**Given** that I am on the 'Landing' dashboard, **When** I click on the 'Settings' button on the top right hand corner and select 'Manage Shared Accounts', **Then** I can view, manage, add, and remove my bank accounts as per the CDR (Open Banking) Rules requirements.
Priority	Must have *(restricted by limitations in regulations for scope of project)*

ID	User Story 7
Feature	Link my personal bank account(s)
Description	**As a user**, I want to be able to link my personal bank accounts, **So that** I can view all my transactions and add all shared expenditures from my accounts to the shared transaction history page.
Flow	**Given** that I am on the 'Manage Shared Accounts' page, **When** I click on the 'Add Account' button, **Then** I am redirected to the consent process to authorise my bank to share my data for my additional account(s) as per the CDR (Open Banking) Rules requirements.
Priority	Must have *(restricted by limitations in regulations for scope of project)*

ID	User Story 8
Feature	View shared transaction history
Description	**As a group member**, I want to be able to view a list of shared expenditures, **So that** I track shared payments that relate to my group.
Flow	**Given** that I am on the 'Group Overview' dashboard, **When** I click on the 'View Shared Transaction History' button, **Then** I will be redirected to a page where I can view the shared transactions, with the following information: • Name of transaction; • Total amount of transaction; • Payer of transaction; • Amount owing of transaction.
Priority	Must have

ID	User Story 9
Feature	Review a shared bill of an expenditure
Description	**As a group member**, I want to be able to review a shared expenditure, **So that** I can ensure that it correctly reflects the financial exchange that it represents.

Flow	**Given** that I am on the 'Shared Transaction History' page, **When** I click on the 'Review' button, **Then** I will be directed to a review page where I can:
	• View the transaction information, including:
	  • Name of transaction
	  • Total amount of transaction
	  • Payer of transaction
	  • Amount owing of transaction
	• Add notes/reminders to the transaction;
	• Change the breakdown of the payment by specifying an amount or percentage per member;
	• Dispute the payment.

Priority	Must have

ID	User Story 10
Feature	Dispute a shared bill of an expenditure
Description	**As a group member**, I want to be able to dispute a bill with other users, **So that** I can correct a bill that I perceive to be incorrectly added.
Flow	**Given** that I am on the 'Transaction Review' page, **When** I click on the 'Dispute' button, **Then** I will be directed to a page where I can enter a reason for disputing, and raise the dispute with other group members.
Priority	Should have

ID	User Story 11
Feature	Change the cost breakdown of an expenditure
Description	**As a group member**, I want to be able to change the breakdown of an expenditure, **So that** expenditures are accurately distributed.
Flow	**Given** that I am on the 'Transaction Review' page, **When** I click on the 'Change Breakdown' button, **Then** I will be directed to a dispute page where I can enter a reason for disputing.
Priority	Could have

ID	User Story 12
Feature	Approve a disputed transaction
Description	**As a group member**, I want to approve/disapprove a disputed or adjusted transaction, **So that** I can make sure that all payments are distributed fairly among shared users.
Flow	**Given** that I am on the 'Shared Transaction History' page, **When** I click on a transaction with a noted dispute awaiting my approval, **Then** I will be taken to the 'Review Transaction' page which will display an option to accept or decline the dispute.
Priority	Should have

ID	User Story 13
Feature	View the status of a disputed transaction
Description	**As a group member**, I want to be able to view the status of the disputed transaction, **So that** I know whether a dispute has been resolved.
Flow	**Given** that I am on the 'Shared Transaction History' or 'Review Transaction' page, **When** a transaction listed is awaiting an approval after a dispute/adjustment, **Then** a 'Pending Approval' status will be shown.
Priority	Should have

| ID	User Story 14
Feature	Manually add a cash expense to the group expenditures
Description	**As a group member**, I want to be able to manually add expenses that I have made on behalf of the other users to the shared transaction list, **So that** I can get reimbursement for a purchase that I have made.
Flow	**Given** I am on the 'Shared Transaction History' page, **When** I click on the 'Add Transaction' button, and the 'Cash' option, **Then** I will be redirected to a page where I can enter the details of the expense including: • Name of expenditure; • Amount; • Date; • Photo evidence (of purchase and/or receipt).
Priority	Should have

ID	User Story 15
Feature	Quick add a transaction from my personal banking transactions to the group expenditures
Description	**As a group member**, I want to be able to add a shared expense that I made with my bank account to the shared transaction list, **So that** I can get reimbursement for a purchase that I have made.
Flow	**Given** I am on the 'Shared Transaction History' page, **When** I click on the 'Add Transaction' button, and the 'Bank' option, **Then** I will be redirected to a list of transactions I have made as per my connected bank records, and I can choose specific transactions to be added to the shared transactions list using the '+' button.
Priority	Must have

| ID	User Story 16
Feature	Viewing analytics regarding spending trends and the contributions of each individual through graph representations on the 'Analytics' page.
Description	**As a group member**, I want to be able to view clear and helpful analytics showing spending trends within my group, **So that** I can track and better budget over a time period.
Flow	**Given** that I am on the dashboard page, **When** I click on the 'View Analytics' button, **Then** I will be redirected to a page where a visual representation of my shared transactions and other helpful analytics will be displayed.
Priority	Should have

| ID	User Story 17
Feature	Validating cash purchases using a scanning feature
Description	**As a group member**, I want to be able to use my phone to scan a receipt for a purchase made with cash, **So that** it is included on the shared transactions page.
Flow	**Given** that I am adding a transaction that I made by cash, **When** I click on the 'Add Transaction' button, and the 'Cash' option, **Then** I will be redirected to a page where I can use my phone camera to scan a receipt and have it interpreted and included as a 'Shared Transaction', with important information (Name, Amount) extracted automatically.
Priority	Won't have

SEQUENCE DIAGRAM

Epic Story 1: As a user, I want to be able to create an account or login to an existing account to access this application.
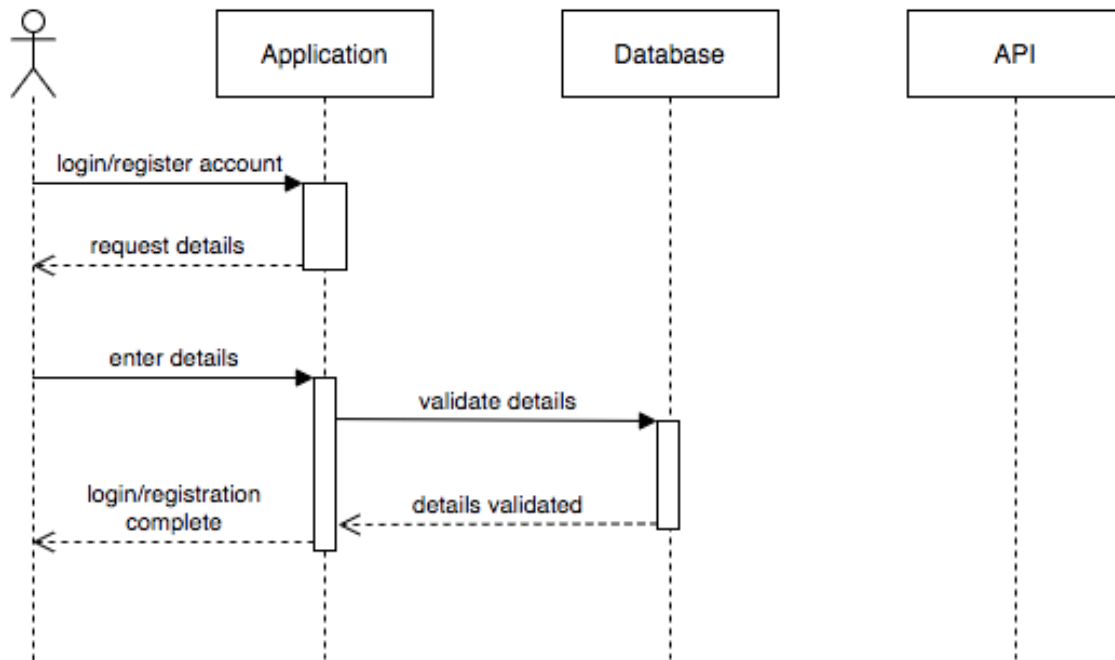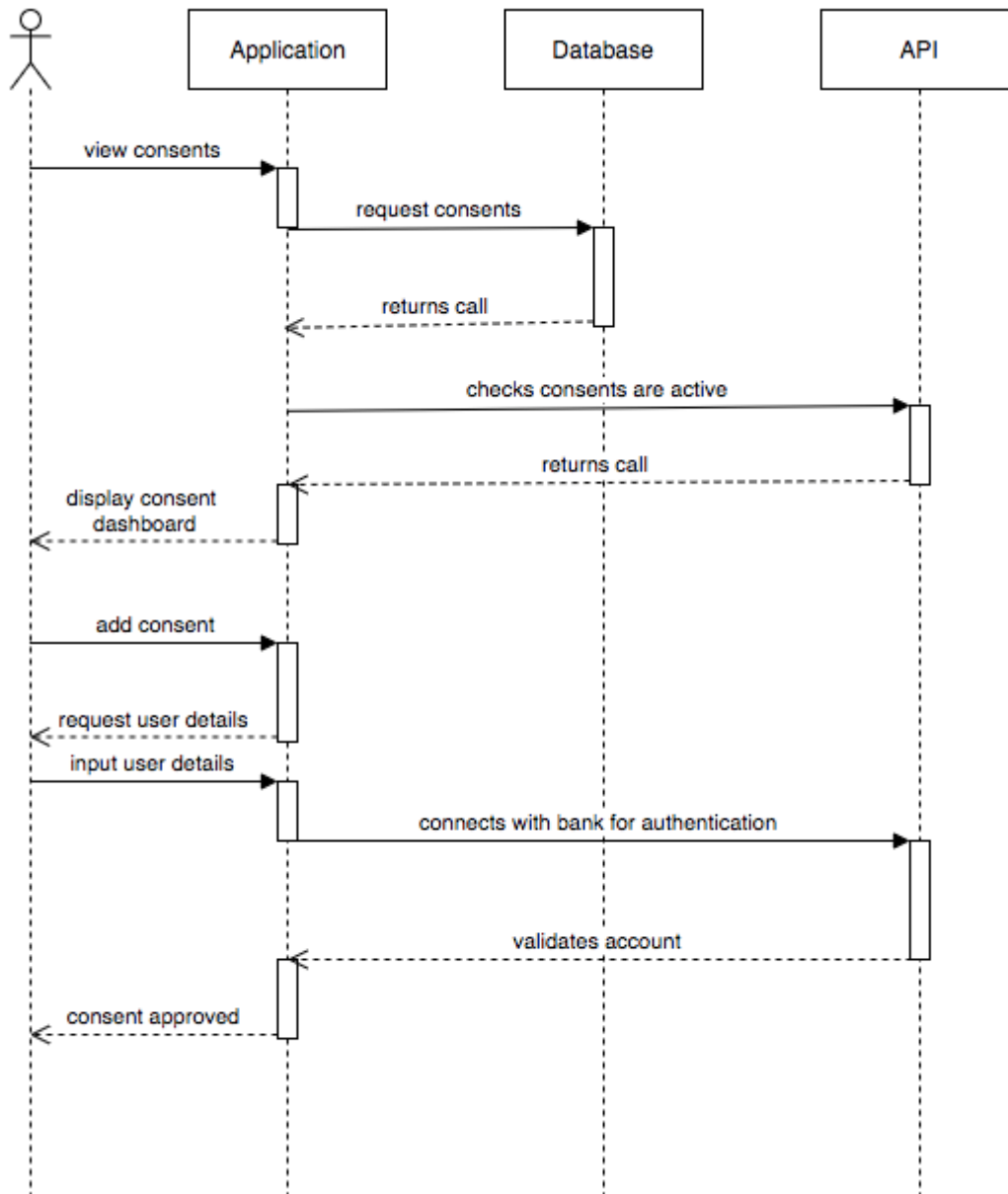
Epic Story 2: As a user, I want to be able to create new groups and have the ability to add or remove members from the group and be notified of updates to existing groups (such as email notifications if I am added to a new group).
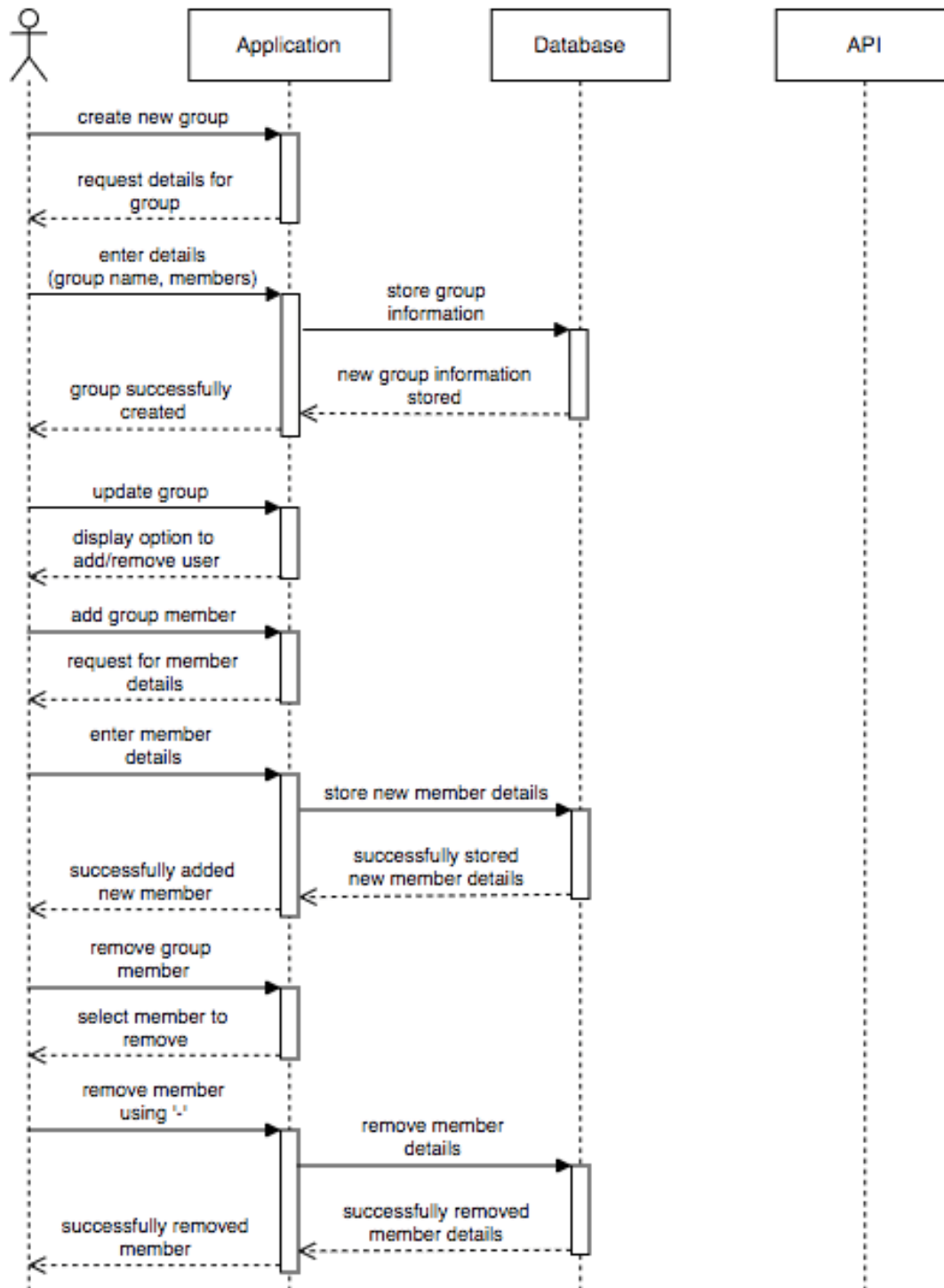
Notification of updates to existing group (continuation epic story 2)
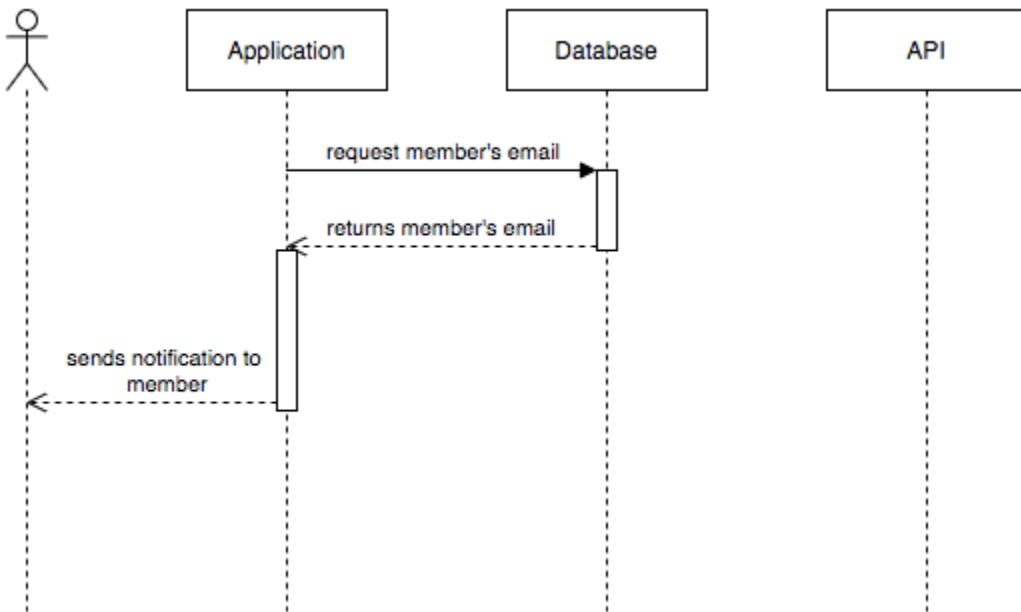
Epic Story 3: As a user, I want to be able to link my personal bank accounts to this application and review or update consents for my bank to share my account information.
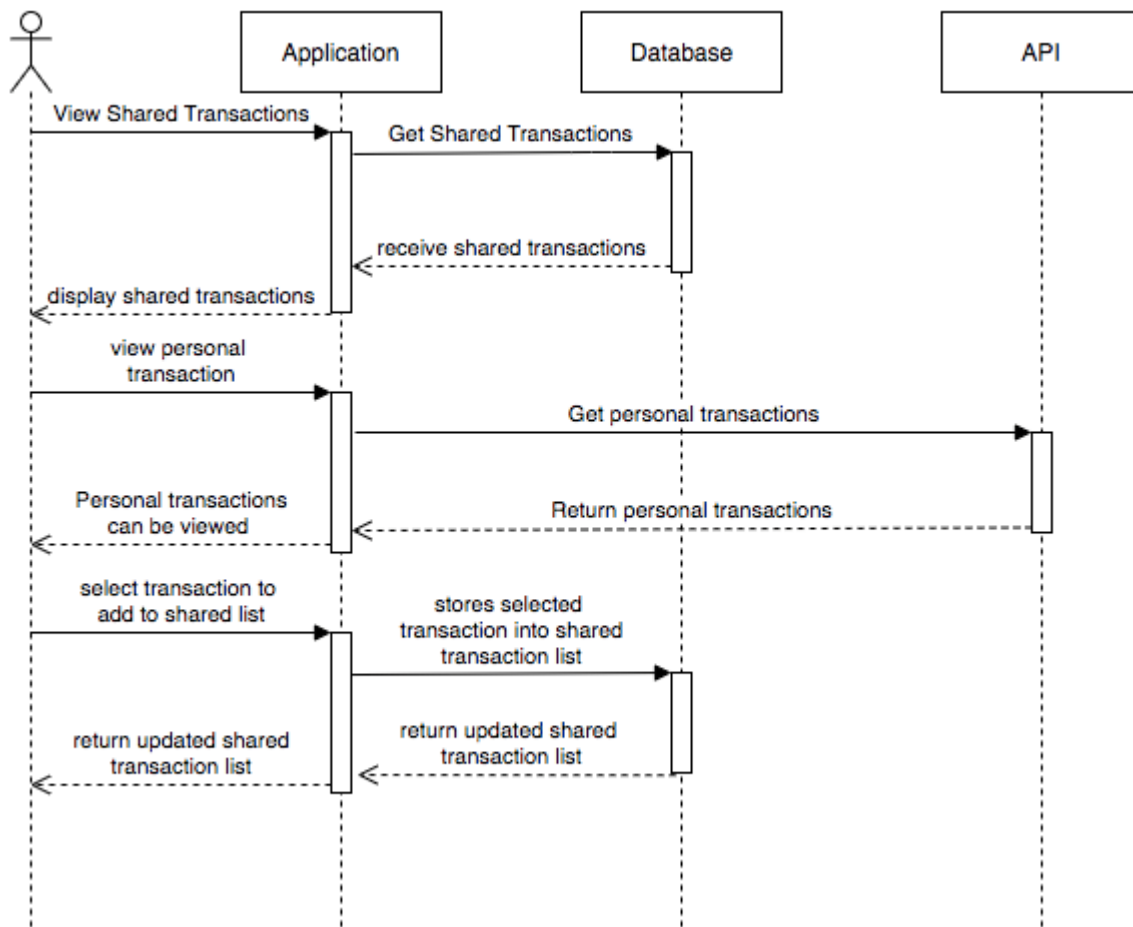
Epic Story 4: As a user, I want to be able to view the shared transactions and have the ability to review or dispute payments with other members of the group.
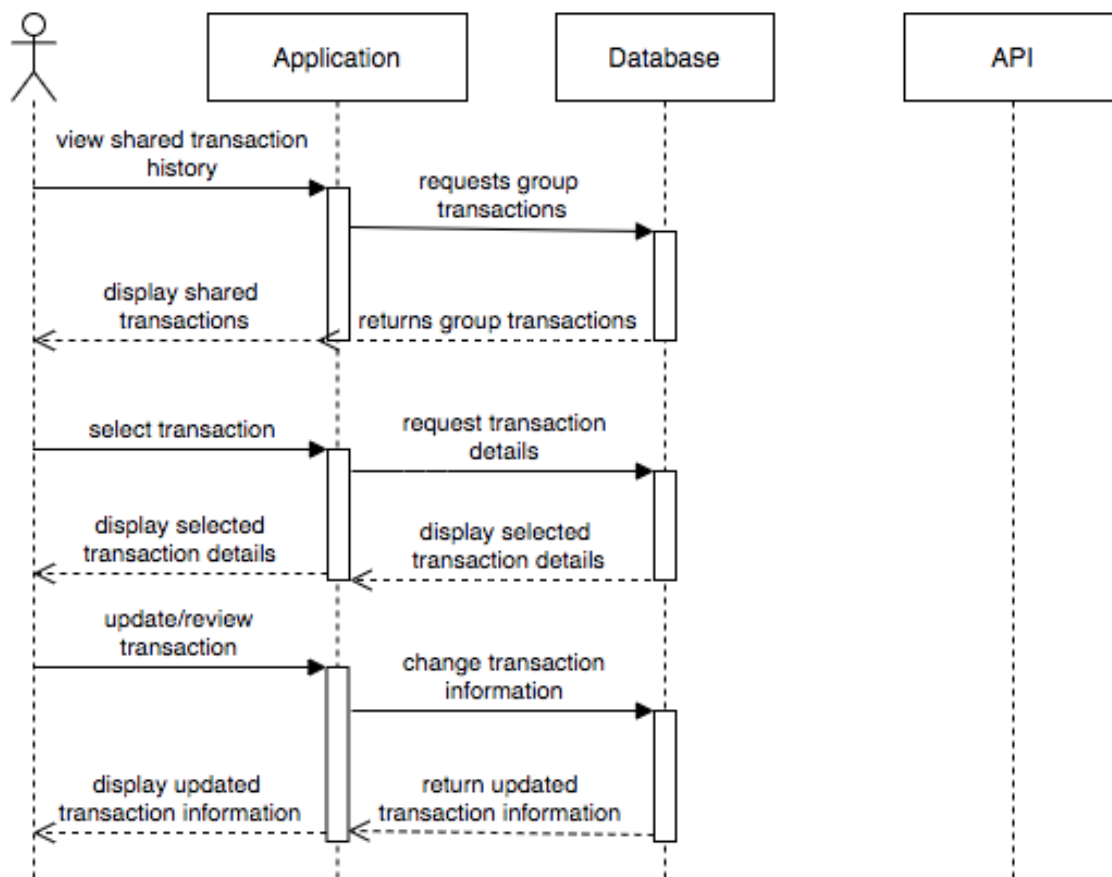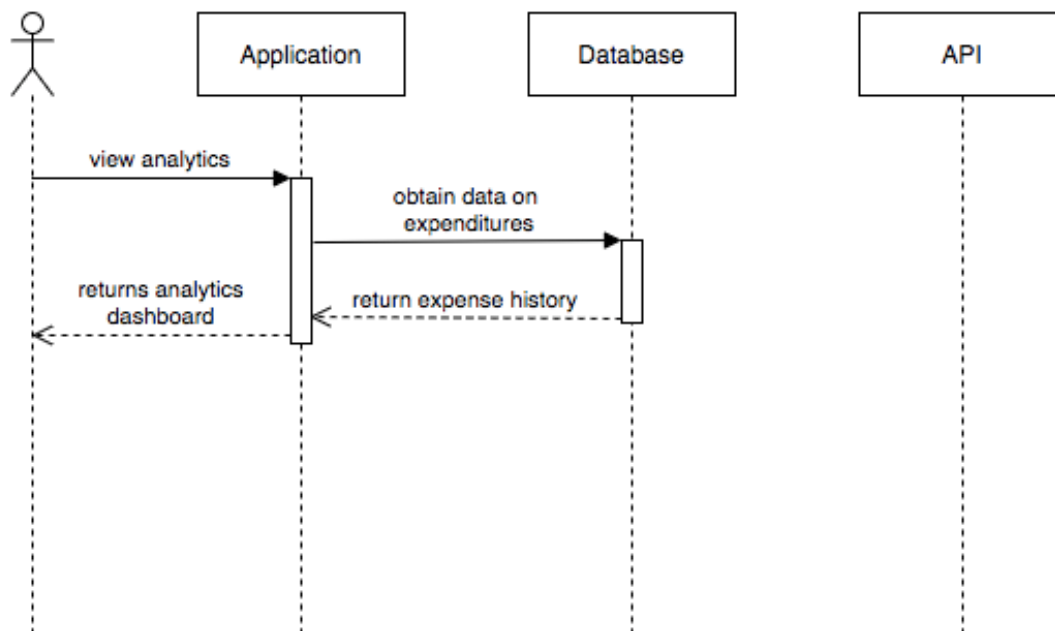
Epic Story 5: As a user, I want to be able to add cash or card transactions to a shared transactions list.
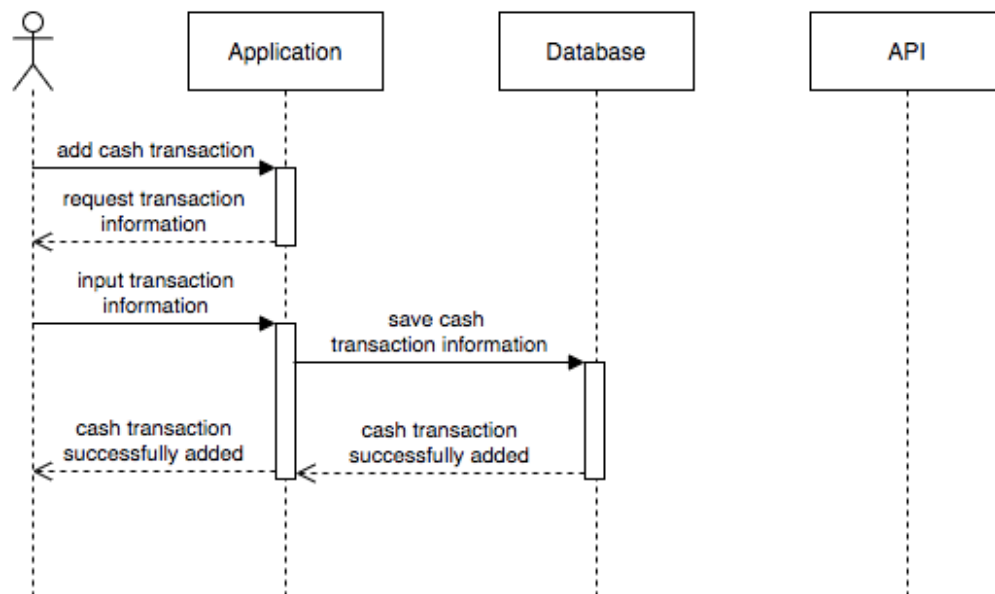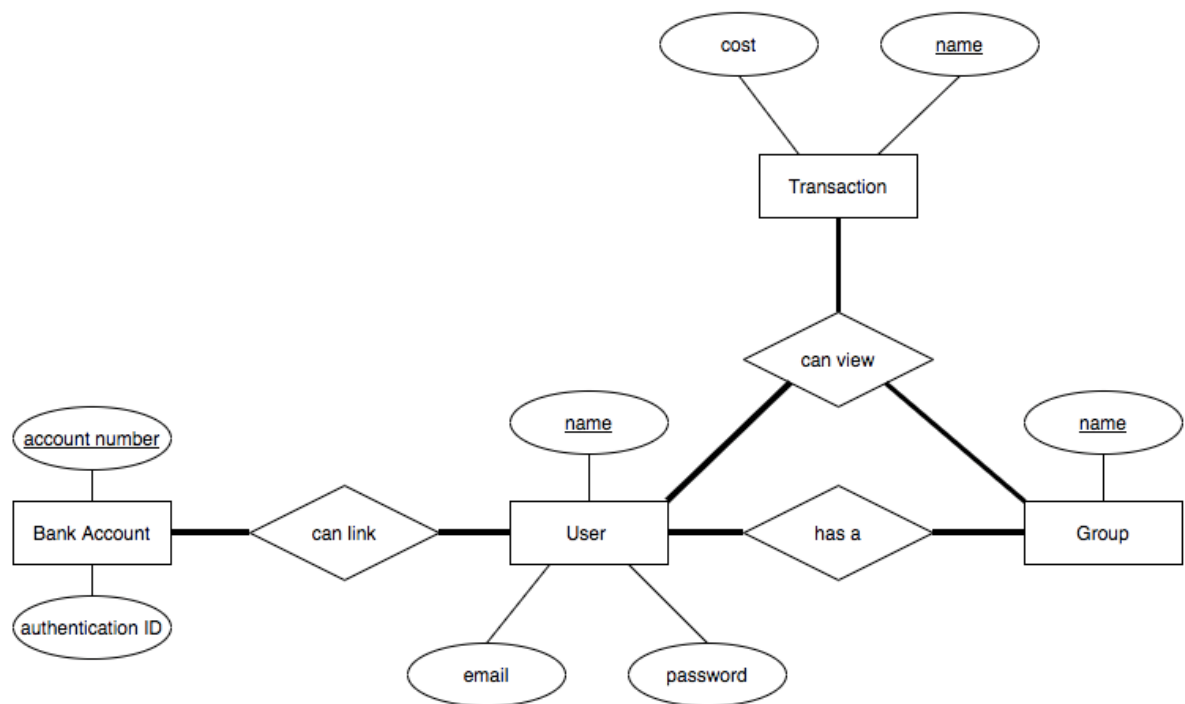

Adding card transactions

Epic Story 6: As a user, I want to be able to view analytics to see my personal expenses and group expenses.

ER DIAGRAM

UML DIAGRAM

Back end System

- transactions: Transaction Database
- users: list<user>
- groups: list<group>

+ add_trans_recipients(transaction, user): void
+ rem_trans_recipients(transaction, user): void
+ upd_trans_ratio(transaction, user, float): void
+ add_transaction(user, float, string, date): void
+ signup(string, string): user
+ authenticate(string, string): bool

Banking API

+ getTransactions(bank_account): JSON object

Transaction Database

- transactions: list<transaction>

Transaction

- name: string
- paid_by: user
- amount: float
- paid_to: string
- paid_date: date
- group: group

Group

- name: string
- members: list<user>

User

- name: string
- email:string
- password: string
- group: list<user>
- bank_account: bank_account
- group: group

Bank Account

- account_number:string
- authentication_ID: string