



UniPay

Product Requirements Document

Team Yao and "Friends"

Yuan Yao
Chandler Forrest
Jake Bliss
Vamsi Kalidindi
Alex Ngo

Table of Contents

Introduction	2
Problem	2
Solution	3
Assumptions	3
Innovation and Science	4
Web App	4
NFC	4
Machine Learning	4
System Architecture Overview	5
Walkthrough	5
User Interaction and Design	6
Landing Page	6
Login Page	6
User Page	7
User Stories	8
Students	8
Administrators	11
Browser Clients	11
Code Samples	12
Login Page Rendering (Frontend)	12
Posting an Earning (Backend)	13
System Model	14
Sequence Diagram for New User Login	14
Sequence Diagram for NFC Charge	15
Appendix	16
Tech Stack	16
Other Applications	16

Introduction

Problem

A significant portion of becoming an adult in modern America is learning financial responsibility. One of the hurdles that young adults face while focusing on school is taking control and managing their money. It becomes difficult to live within your means when your school expenses are abstracted away online. The ideal solution would provide useful statistics to students based on their spending habits, show exactly where their money is going while being able to deposit and withdraw money instantly from virtually any source.

To provide context, one of the first things UCSB students learn during their freshman orientation is how to properly pay their dues through UCSB's online payment portal, Billing Accounts Receivable Collection Unit (BARC). BARC keeps track of each student's tuition, on-campus rent, health insurance, dining commons meal plans, residence hall printing, lab materials, and other seemingly random fees. It is confusing to see breakdowns of your feeds, hard to get money in and out and provides a frustrating user experience.

BARC's biggest issue is its inability to show users exactly how money is being spent on campus. As students, it's very crucial to filter where money is being spent so that you can save money and budget appropriately. On BARC, all your charges are just pushed into a big list, which cannot be manipulated or filtered. Furthermore, you can't see how much is spent for a given month or period of time. This is an issue because students should be able to see when they're spending the most money, or for any given quarter that they are enrolled in. If they see that their expenses are higher for a given quarter, they can choose to cut those costs during the following quarter.

Another one of BARC's biggest issues is the way money flows in and out of the system. In order to put money into your BARC account. There are three options - credit card, electronic checks, or international payment. Paying with credit card is ridiculous because it charges a 2.75% fee for the transaction. If you are paying in-state tuition for a year, that's an extra \$300. International payment makes you complete the process inside another application instead of handling it natively. This leaves the only sensible way to pay is through an electronic check which makes you enter your full banking information every time you try to complete a transaction. Finally, in order to pay for a due, you actually check it out into a shopping basket and pay for it using one of the three options. It gives you the impression that you are given a choice whether to pay or not. When you have a positive balance - which if you can figure it out, means the school actually owes you money - they mail you a check. A large improvement we aim to create is making the process of taking money in and out of BARC as simple as possible while generating a more sincere way of paying for dues.

Furthermore, BARC doesn't let you export any of this data into a more viewable format, such as a CSV file. If the user wants to further manipulate this data in Excel or any other software, it's not possible. This is something worth tackling because students shouldn't be solely dependent on BARC's bill tracker to see how much they're spending. Within BARC's financial expenses, it's also hard to keep track of which expense falls into a certain category, such as a restaurant expense, or quarterly gym fee. Everything is added together as a whole bill, which doesn't help filter where money is going. Therefore, one of our biggest priorities to combat this issue is to provide analytics with the data being saved into the new payment application

Solution

Our System provides a unified, easy to use platform for all student finances. To accomplish this, we draw financial information from Workday Cloud Platform, on-campus purchases, and online purchases related the university. We then analyze this information, generate informative analytics and display them to the user.

We also want to make payments for on-campus charges as simple as possible. To integrate on-campus purchases into our web app, we provide an Android app which allows a student to make payments at on-campus stores via Near Field Communication or with a QR code. If you are purchasing something like a cup of coffee, the entire interaction should take less than 20 seconds. We aim to do this by creating a simple companion app that uses your phone as the payment system. Any purchase made on campus would be reflected for students to see and reflect upon at a later time.

Our system will make use of artificial intelligence to track student's financial habits and provide suggestions. These suggestions are intended to help guide the student's spending so they can budget appropriately and maintain healthy financial habits.

Assumptions

- The student initially has an account on Workday student
- The student is connected to the internet for NFC payments
- Administrators have integrated UniPay API calls to post a student's charges and earnings

Innovation and Science

Web App

As of the web app, we are using the combination of ReactJS + NodeJS. By defining a single language for the front end and back end, all five of our team members can code review each other's code and learn more than just our own parts. In addition, NodeJS backend provides a significant amount of community contributed frameworks. As for our plan right, we are using Express and MongoDB for our backend routing and storing services.

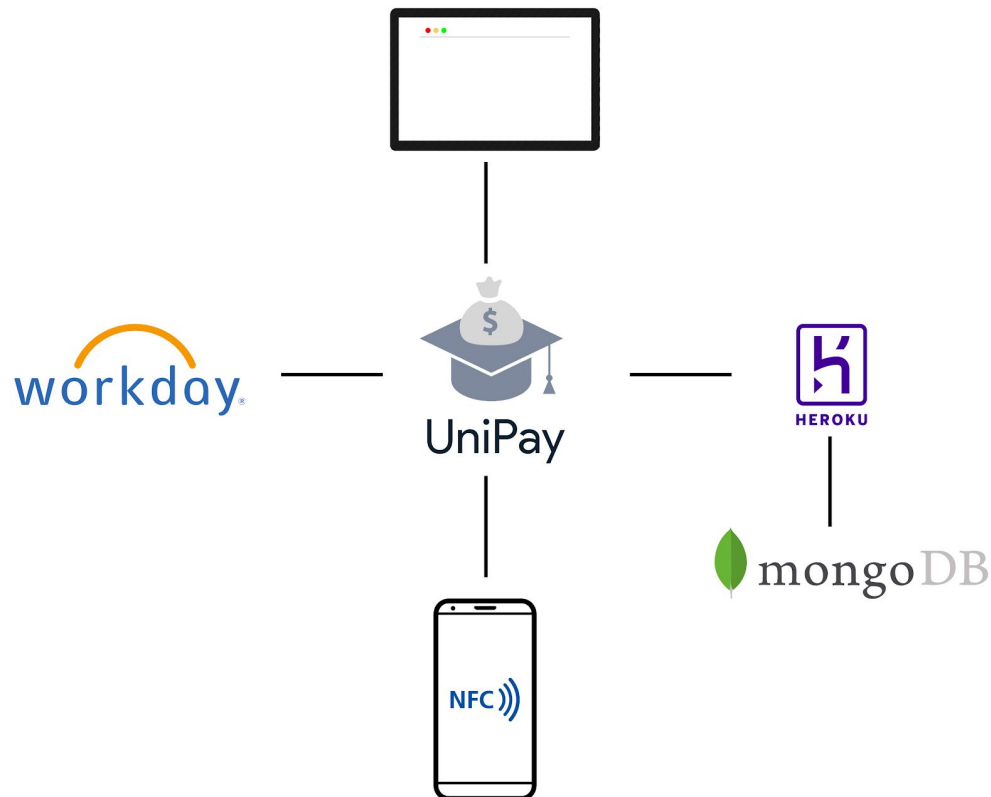
NFC

One of our innovation about our product is the NFC/QR payments on campus. Since our goal is to unify and simplify the entire money flow system for universities, it makes sense for us to figure out a simple way where students can make purchases on campus directly from their UniPay account with a snap of a finger. To accomplish this, we need to develop a companion mobile App for students. We will have NFC stands set up at the cashier counter. When students make purchases, they can easily tap their phone to the stands and the payments will be automatically billed to the students' account. During the research process, we found out that Android Phone can simulate NFC signals. So this is a great start for us to play around with this technology where we would have two Android device, one emitting signals, mimicking the NFC stands at the cashier counter, and another phone receiving the signal, mimicking when students use their phones to pay. This requires utilizing the Connectivity library coming from the Android framework.

Machine Learning

In the future, after we have implemented everything nearly all functionalities of both the web and mobile app, we would like to explore the possibilities with machine learning to generate suggestions on how to improve students' spending habits. We could also do some data manipulation and comparisons to let the student have a better idea of how much they are spending compared to their peers.

System Architecture Overview



Walkthrough

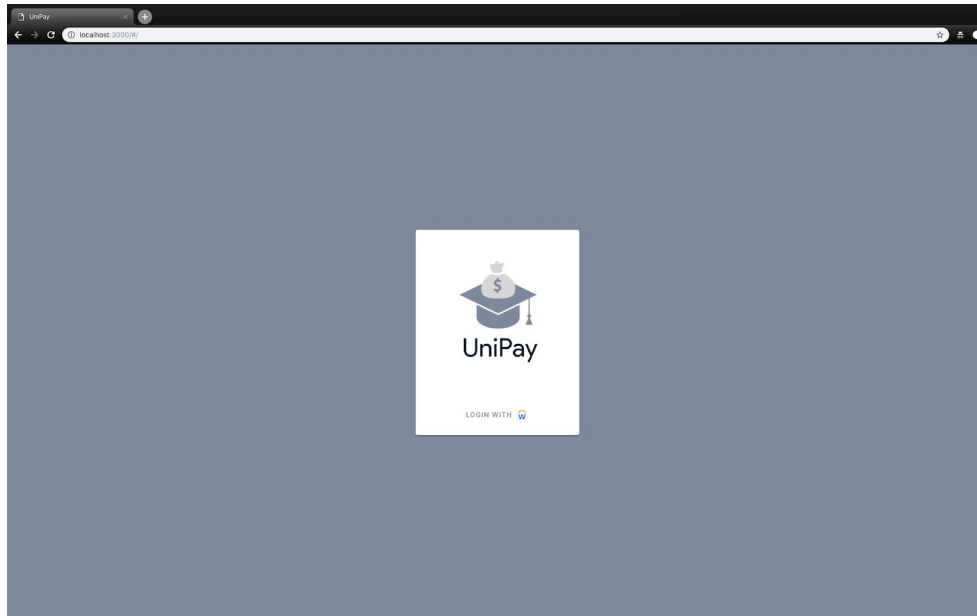
Students start by logging in through Workday which properly authenticates them, and retrieves their data if they are a new user. From there they can view their charges and earnings through the web app which is hosted on Heroku which is running MongoDB as its database.

Businesses on campus can choose to incorporate NFC capabilities which allow students to simply tap their phone to make any on-campus purchases. These purchases would be POSTed to UniPay and tossed into our backend for safe keeping.

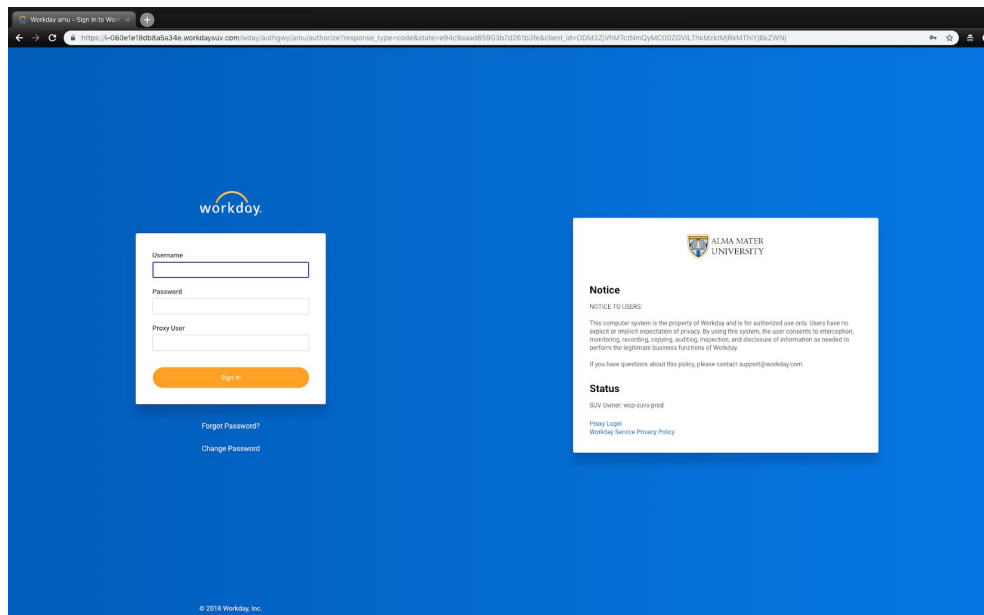
Additionally, any campus entity can choose to POST a student's charges and earnings through UniPay. This allows students to pay for everything on campus in one easy application.

User Interaction and Design

Landing Page



Login Page



User Page

The screenshot shows a web browser window with the URL localhost:3000/#/UserPage. The page is titled 'UniPay' and greets the user 'Hello, Abigail'. It features two main sections: 'Recent Charges' and 'Recent Earnings'. Each section contains a grid of cards representing individual transactions. The 'Recent Charges' section has 7 cards with amounts ranging from \$75 to \$3500, all categorized as 'school_charge'. The 'Recent Earnings' section has 7 cards with amounts ranging from \$110 to \$392, all categorized as 'job_deposit'. Each card includes a date, a descriptor, and a star icon.

Amount	Category	Date	Descriptor
\$ 2800	school_charge	Thu Aug 16 2018	12 Meal Plan
\$ 3500	school_charge	Mon Dec 18 2017	19 Meal Plan
\$ 186	school_charge	Mon Dec 18 2017	Student Health Service Fee
\$ 186	school_charge	Tue Jan 09 2018	Student Health Service Fee
\$ 75	school_charge	Thu Aug 16 2018	Lab Fee: PHYS 202A-L1 - General Physics I
\$ 90	school_charge	Mon Dec 18 2017	Technology Fee
\$ 110	job_deposit	Tue Sep 25 2018	
\$ 110	job_deposit	Mon Apr 30 2018	
\$ 270	job_deposit	Tue Nov 20 2018	
\$ 392	job_deposit	Tue Oct 09 2018	
\$ 308	job_deposit	Mon May 14 2018	
\$ 140	job_deposit	Mon Jun 25 2018	

User Stories

Students

- 1) As a student, I want to be able to login to the website, so that I can access my financial information.
Acceptance Test: After a successful login, the user is redirected to the home page.
Trello: <https://trello.com/c/PRlrosBO/15-investigate-create-login-page>
Github: [Acceptance Tests](#)
- 2) As a student, I want to be able to log out of the website, so that I can ensure that no one can access my information through the web application after.
Acceptance Test: After clicking the logout button, the web application redirects to the login page and any attempts at accessing internal pages are blocked.
Trello: <https://trello.com/c/a9xNOn0Z/58-logout-of-web-app>
Github: [Acceptance Tests](#)
- 3) As a student, I want to be able to view my profile page, so that I can see my own information.
Acceptance Test: After clicking on the profile icon, the page redirects correctly and displays a card with the user's information.
Trello: <https://trello.com/c/x6hSS7nz/17-display-basic-info-on-web-app-based-on-id>
- 4) As a student, I want to be able to view all my charges/earnings, so that I can get a snapshot of all my transactions.
Acceptance Test: After hitting the view all button for charges and earnings, the webpage redirects and a page with a list of charges/earnings is displayed to the user.
Trello: <https://trello.com/c/stxPvtai/59-view-all-charges-earnings>
- 5) As a student, I want to be able to view my financial history, so that I can keep track of my finances.
Acceptance Test: After clicking on the transactions tab, a redirect should occur and a page showing a table of past transactions should appear.
Trello: <https://trello.com/c/hkEBitos/18-show-financial-history>
- 6) As a student, I want to be able to check my current balance, so that I know how much money I have.
Acceptance Test: After logging into the main page, a card should show up displaying how much money is currently in the account.
Trello: <https://trello.com/c/gOErui1q/19-show-current-balance-balance-due>

- 7) As a student, I want to be able to see my transactions during a certain month, so that I can be better informed of how I spend my money through certain periods of time.
Acceptance Test: After clicking on a drop-down menu of month names and clicking on a specific month, the only transactions that appear are from that month.
Trello: <https://trello.com/c/Vn4Jgdl1/63-view-charges-earnings-associated-with-a-specific-month>
- 8) As a user, I want to be able to see analytics about my spending habits, so that I will be informed of how I'm spending my money where and when.
Acceptance Test: After clicking on the analytics page, information is shown based on the user's earnings and spending habits (e.g. how much you spent on X, Y, Z, how much you spent vs other students)
Trello: <https://trello.com/c/NVh8o4mw/60-basic-analytics>
- 9) As a student, I want to be able to easily make purchases around campus so that I can save time and not have to worry about immediate payment.
Acceptance Test: After opening the mobile application and scanning NFC, it appears in the database as a charge.
Trello: <https://trello.com/c/hkEBitos/18-show-financial-history>
- 10) As a student, I want to be able to generate a pie graph based off my finances, so I can easily see where expenses are being made.
Acceptance Test: On home page, a pie graph is generated that is split via type (e.g. meal plans, gym, restaurants, lab fees etc.)
Trello: <https://trello.com/c/c0N6py0l/62-generate-pie-graphs-based-on-user-data>
- 11) As a student, I want to be able to generate a PDF to view a specific billing statement so I don't need to access the online platform in order to view my previous bills.
Acceptance Test: On local device, a pdf is generated that shows a specific billing statement that a user can access without having to log onto UniPay.
Trello: <https://trello.com/c/9DUUfdku>
- 12) As a user, I want to be able to change my user information, so I change update my user information in the case of things like phone number/email changes.
Acceptance Test: After changing a field in the change user settings page, when navigating to the profile page, the updated information is correctly displayed.
Trello: <https://trello.com/c/EaZ81wc6/61-update-user-information>

13) As a user, I want to be alerted when there is a upcoming bill that needs to be paid, so that I can pay my bills on time.

Acceptance Test: Create a charge that is in the future and check if a notification is displayed on the dashboard warning the user that there is an upcoming bill.

Trello: <https://trello.com/c/kLwyMDpD/64-upcoming-bill-alert>

14) As a student, I want to be able to pay an upcoming bill through the app because it is very convenient to be able to pay my bill in a finance app.

Acceptance Test: After paying for a bill, the bill is correctly marked as paid in the database and a transaction should the paid bill should pop up.

Trello: <https://trello.com/c/0JguRlxw/65-pay-for-a-bill>

15) As a student, I should not be able to create a budget that exceeds my earnings, because I would fall into debt

Acceptance Test: Create a budget that exceeds a user's earnings, it should not allow the user to create the budget and display a warning.

Trello: <https://trello.com/c/ceedgpla/66-budgeting>

16) As a student, I want to be warned when I am spending outside of my budget, so that I can manage my finances better.

Acceptance Test: Based on a user's budget, if a charges for a certain category goes over the budget the user set, a notification should appear telling the user they are spending too much money.

Trello: <https://trello.com/c/WdOrs691/68-over-budget-warning>

17) As a student, I want to be able to subscribe for periodic updates on my account, so I can be regularly updated without having to log into the web app all the time.

Acceptance Test: Simulate sending an email to the user if they are subscribed based on the email in their profile, check that the user actually received an email.

Trello: <https://trello.com/c/YFOJjvEZ/67-periodic-updates-to-user>

Administrators

- 18) As an admin, I want to be able to post a charge/earning associated with a student to their account database, so the student can keep track of their finances with valid data.
Acceptance Test: After a post request to the backend api, a response is received with a valid JSON object.

Trello: <https://trello.com/c/MCyYAJKW/9-charge-route>

Github: <https://github.com/why Yao/UniPay/pull/14>

- 19) As an admin, I want to be able to create a new user in the database, so that the student can begin to use the website and data associated with them can be stored.

Acceptance Test: After a post request to the backend API, a response is received with a valid JSON object.

Trello: <https://trello.com/c/rW7ZlohF/27-create-student-route>

Github: <https://github.com/why Yao/UniPay/pull/15>

Browser Clients

- 20) As a browser client, I want to be able to authenticate a user through Workday, so that they may access the website afterward.

Acceptance Test: After workday authentication redirects back, a valid json web token is issued (Issued in the URL).

Trello: <https://trello.com/c/23iGe1ZG/42-authentication-for-login>

Code Samples

Login Page Rendering (Frontend)

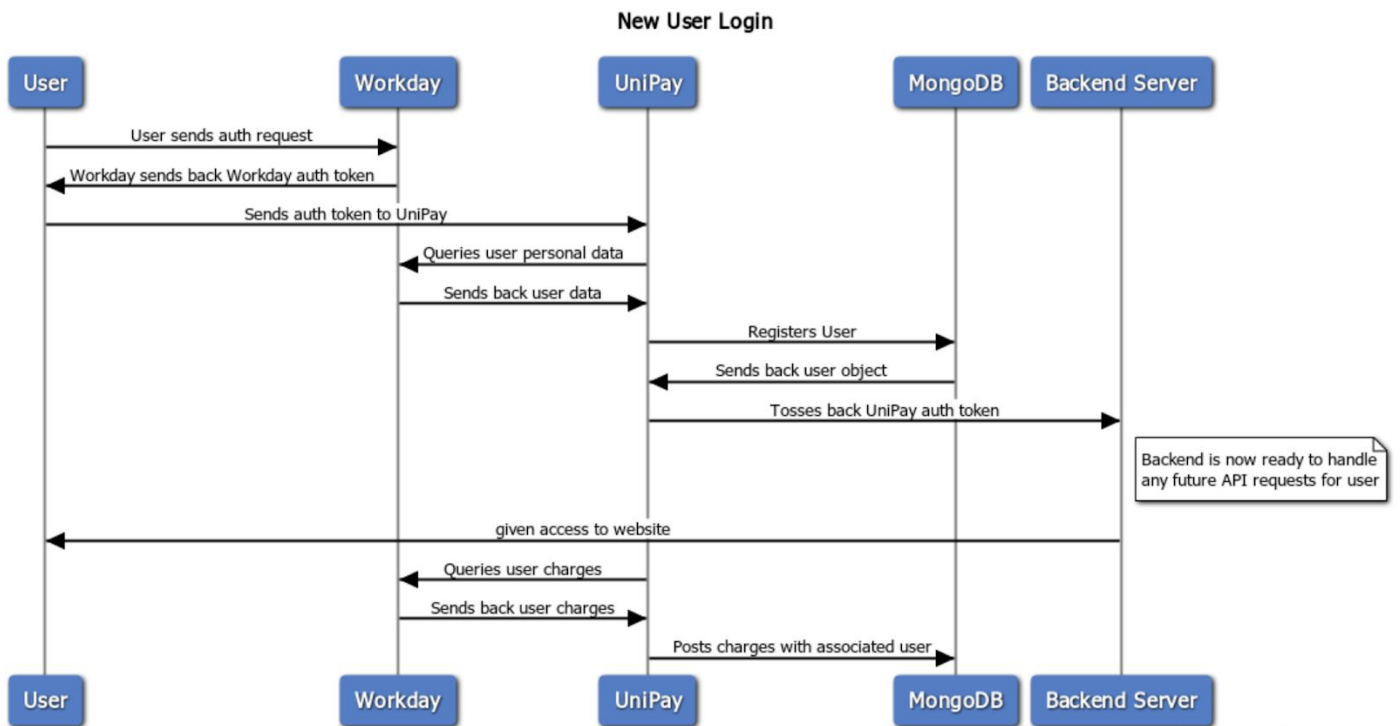
```
render() {
  return (
    <div>
      <Card className="login-page" style={{ width: "20rem" }}>
        <CardMedia
          square
          style={{
            backgroundImage: "url(/unipay_with_text.png)"
          }}
        />
        <Button
          style={{ height: "5rem", fontSize: "14px", paddingRight: "5px" }}
          onClick={this.handleClick}
        >
          Login with
          <ButtonIcon
            style={{ paddingLeft: "10px" }}
            icon="/workdaylogo.png"
            iconOptions={{ strategy: "url" }}
          />
        </Button>
      </Card>
    </div>
  );
}
```

Posting an Earning (Backend)

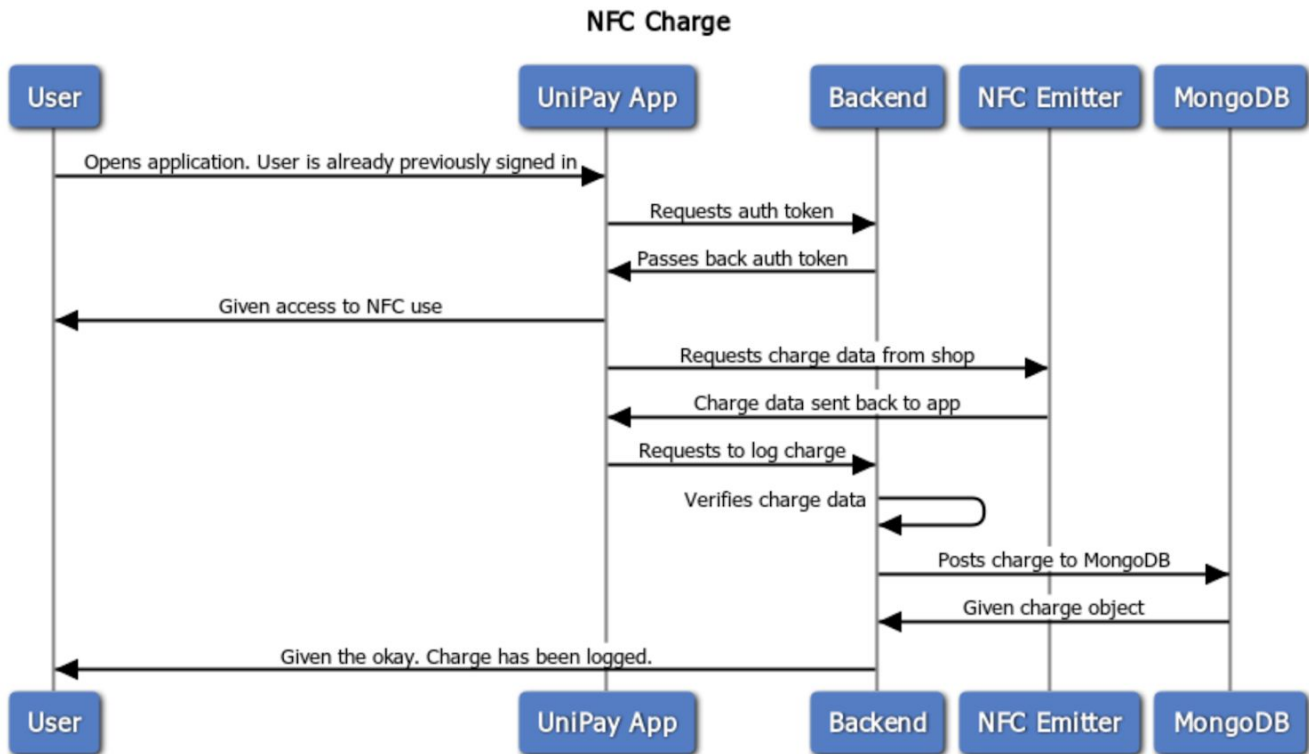
```
router.post('/', async (req, res) => {
  const { error } = validate(req.body);
  if (error) return res.status(400).send(error.details[0].message);
  let earning = new Earning({
    amount: req.body.amount,
    type: req.body.type,
    student_id: ObjectId(req.body.student_id)
  });
  earning = await earning.save();
  if (!earning) return res.status(404).send(earning.error);
  res.send(earning);
});
```

System Model

Sequence Diagram for New User Login



Sequence Diagram for NFC Charge



Appendix

Tech Stack

- Node.js for web app handling
 - Jest used for unit and integration testing
- Express.js for API's
 - Postman used for testing API calls
- React.js for frontend UX
 - Jest also used as a unit testing framework
 - Material.io for styling
- Workday Student API for student creation
 - Used in combination with JWT for UniPay authentication
- MongoDB for database
 - MongoDB compass for testing DB locally
- Android/Java for NFC mobile charges

Other Applications

- GitHub for version control
- Heroku for web hosting
- Trello for ticketing
- Zoom for mentor meetings
- Google Hangouts for standups
- Slack for chatting
- Google Docs for documentation