

Group Report 2



Bravo 6

By Caleb bowden, Annum Mubarik, Raul Rahaman, Samuel Janney, Nathan Walsh
08/04/2022

Introduction	3
Merits and limitations of technology	3
Problems encountered and solutions proposed	4
Team roles and skills	5
Class Diagram	7
Figure 1 - Uni Pay class diagram.	7
Sequence Diagram	8
Figure 2 - Uni Pay sequence diagram	8
Product management plan - Waterfall	8
Conclusion	9
Bibliography	10
Appendices	11
Appendix A: Project Plan	11
Appendix B: Agenda / Meeting Minutes	11
Meeting 4 Agenda:	11
Meeting 4 Minutes:	11
Meeting 5 Agenda:	12
Meeting 5 Minutes:	12
Meeting 6 Agenda:	12
Meeting 6 Minutes:	13
Meeting 7 Agenda:	13
Meeting 7 minutes:	13
Appendix C: Screenshots of Software	14

Introduction

UniPay is a Application designed by students for students which helps track and control their finances to help lift their daily financial burden which increases daily according to Robb (2017). With the “available spend” feature the app will show you how much you have to spend a day by working out your income and outgoing, Students can set their work income and student loans and also their bills. This report will go over issues that were encountered and solutions on how to fix the issue, the pros and cons of the technology and the limitations that occurred during the project, sequence diagrams and the product management plan on how the team organised themselves and distributed the workflow. Developing software using certain technology can have its merits and limitations in which the team had to adapt in order to succeed. There were some problems encountered that disrupted the workflow and the team had to come up with solutions to overcome those issues. Further, there is a class diagram and a sequence diagram alongside the product management plan that helped in the planning and development of the project. The conclusion can be found at the end and it summarises the development process. The references can be found in the bibliography and in the appendices we provide proof of our work, such as the project plan, agendas and the screenshots of the application.

Merits and limitations of technology

Financial limitations - Because the team did not have funds or any way to finance the development of the application the developers could not host a database that would store user data and logs, this meant it was hard to implement a login system and store user data so the team had to use variables to store data. locally in the program using double values. The team also could not afford the needed APIs to connect the application to financial processors and to check for a valid debit / credit card with the correct name so we had to validate the best we could by checking the card / bank accounts format.

Time limitations - The developers had to work within a very small time limit this meant that core features were the top priority like balance system / GUI design / Available spend and Financial tracker. This meant that other features like upcoming payments, login system, recent spending and NFC / RFID payment system could not be completed.

Development Knowledge - during the team meetings we discussed our background knowledge of creating mobile applications and the team members had limited knowledge regarding software development and the programming languages so this meant we would have to learn how to create these applications in turn gives us less time to add features, so the team decided for the demo application we would use C# (Hejlsberg,2008) which is compatible with Windows and a language that follows the object oriented principles which would benefit in the software development.

Language limitations - Developers chose to use C# (Hejlsberg, 2008)because they had previous knowledge of this language and it would be more time efficient to create a demo application instead of learning a whole new language. However this language is not the greatest for mobile application development and it would have been better to go for java (Farrell, 2011)or node JavaScript packages since the lack of experience with these limited us. This meant the code we had written previously could be used as a guide but would have

to be re-written in a different language which would also result in other types of errors which may then result in a design / feature change. Visual Studio (Microsoft, 2022) was the go to IDE for the demo as you can drag and drop objects to easily create a GUI.

RFID Merits - RFID / NFC technology is a new technology which allows data to be sent and then received by a reader. This tech can be used for financial systems for example payments where you can tap your card on a reader for ease of use. Because now most smartphones have these hardware features built into them, payment systems can be developed to use this hardware to send payments, so instead of using your card you can use your mobile phone to pay, making debit / credit cards obsolete.

Student financial merits - As the application is based around students it was important for use to include financial features that will help a student spend. There are 2 balances that appear on the home screen of the app one is the total balance in your account and the second balance is your daily spent allowance. Once you have set up your income and bills the application will work out how much you can spend per day.

Team size limitations - With the competitors we researched, the app was owned by a large scale company, with tens or hundreds of highly skilled developers and designers. With our project, we were limited by the size of our team, and also the programming ability of our team. This meant that we were unable to create an app that is as polished or functional as our competitors.

Github merits - During the programming development, we used github (Thung, 2013) to share files. This was very helpful as it meant we could update the progress we had made using the commit process and also collaborate on the programming.

Visual studio merits - In development we used visual studio as a shell to create the program. This is because it is free, easy to navigate, and reliable, as there weren't really any problems with crashing or corrupting files.

Problems encountered and solutions proposed

During the development process of the application there were a variety of issues that we had to tackle for the process to work correctly.

Double values - The team encountered issues with decimal point numbers eg 12.54 if the user entered a value like this the program would crash with an error. This issue occurred because the application only took integer values which cannot store decimal point values. To fix this the team changed all the integer values to double values which can store decimal point values.

Input validation - Because UniPay handles debit / credit and banking information we must try to validate this information. If validation did not occur a user could enter any random digits names etc. To stop this from happening the input boxes check for specific values and have a

set size for example a debit card number is 16 digits and only contains numeric values. If the user tries to enter an alphabetical value the application will not allow it and that value will not be inputted. Once 16 digits have been entered the user will not be able to input any more.

negative values - To stop a user from entering values that are less than 0 the team developed a check, so every time a user enters a value it checks if this value is a positive number. If this check did not occur there would be calculation errors. The team also discovered an issue with the balance system and the monthly bills. If a user was to enter a there monthly bills and the total amount is greater than what is in your account and your student loans / work income the program would put your balance into negative, to stop this from happening we did a check that gets the users balance and adds it to monthly incomes and then checks if its less than or equal to the month bills. This fix also stopped users from being able to send / withdraw funds they did not have in their balance.

Organisational problems - When it came to submitting our first group report due to organisational problems we forgot to dedicate a member to submit the final report so we didn't know who was going to submit it, due to this multiple members ended up submitting the report. In the future to prevent this happening again we made sure when giving out tasks to dedicate submitting the report to a specific member to make sure only one member submitted it on time.

Team roles and skills

During our first team meeting all of the members discussed what each of our individual skills were and what role each member thought they would be best suited for. The members discussed the best way to choose team roles and decided to vote on who would be best fitted for what role based on their skills and experience.

Team leader -This role involved overseeing the groups progress and dealing with any problems or queries that may come up. The team voted on who would be the team leader based on their communication and decision-making skills and how well they could manage a group of people. Social skills would also be important for this role. We decided that Sam would take on this role as he was confident and trustworthy enough to lead the group. Overall Sam carried out the role quite well and had all the skills needed to lead and guide the group throughout the project.

Developers -This involved developing the app using a programming software. The members voted on this role based on who had the most experience with developing projects and who knew a larger variety of programming languages. This person would need to be able to turn our idea into a working program and take note of any problems or errors that came up during the development. It was decided that Caleb would take on this role as he had the most experience and was confident that he could lead the development of the software demo. Caleb carried out this role very well and managed to create a working program that satisfied all the requirements we needed.

Presenter - The presenter would have to be confident at speaking in front of a large group of people and would be able to easily convey our idea across. Even though all team members

would be presenting we thought it was important to have one person who could lead the presentation and ensure that all points were covered and the audience understood our product. We chose Nathan for this role as he was very confident and was able to speak in a very clear and engaging manner so he would be able to convey our ideas whilst keeping the audience's attention. We were able to give clear presentations and the audience was able to fully understand what we were talking about so this role was fulfilled quite well.

Minute keeper - This role mainly consists of documenting any topics and decisions spoken about during meetings. This person would also need to create agendas to lead the direction of meetings. This person would need to have good organisation and time-management skills as they would be in charge of making sure meetings start on time. They would also need to be a good listener and be able to take in information and compile it into a list for use later. We chose Annum for this role as she had skills and experience for note-taking and was able to take in and understand information to document it. This role was carried out well and we have clear agendas and minutes for all the meetings that can be used for future reference.

Analyst - The analyst of the team would be in charge of gathering ideas and implementing them. They would also be involved in the planning and presentation process. This person would need to be good at researching and would need to have good critical thinking skills. They would need to be able to interpret complex data from research to help the group during the entire process. We chose Raul to do this role as he had all the skills we thought were needed for this role such as analytical and research skills. Overall he carried out the role very well and was able to gather ideas and implement them for the group to use during the planning, development and presentation processes of the project.

Overall, our group was able to quickly and efficiently divide roles between the team members based on our individual skills. We had a wide range of skills between the members so we were able to cover all the skills we needed for the project. The allocation of roles was very effective as each person had all the skills we thought would be needed for that role therefore, all the members were able to carry out their role to a good standard.

Class Diagram

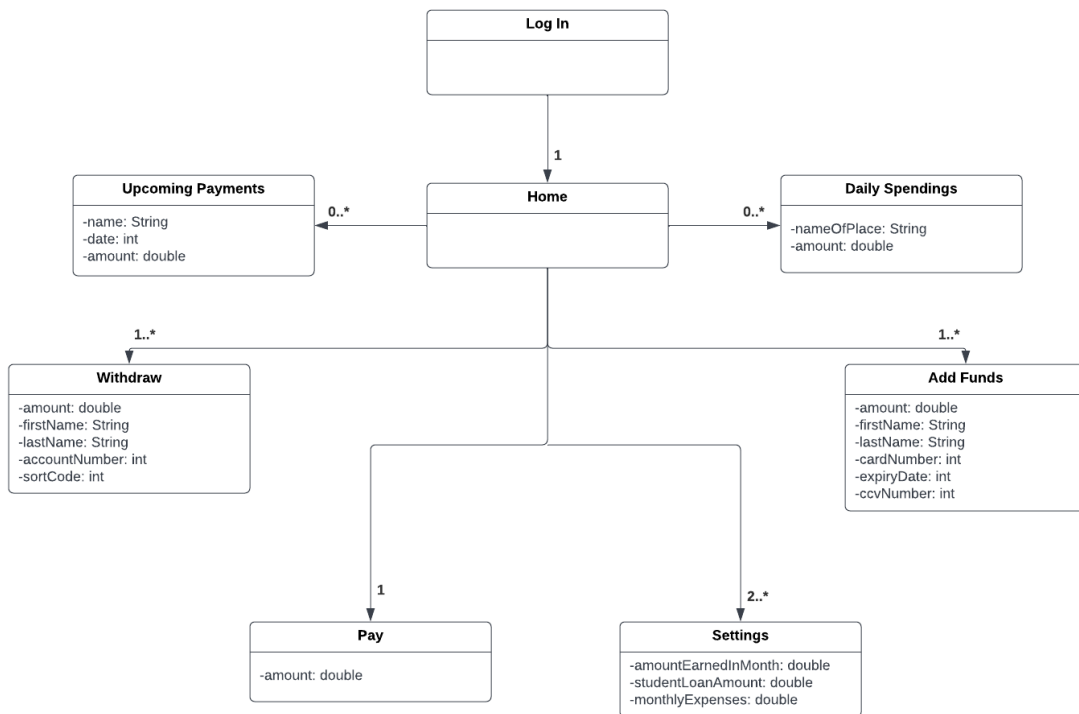


Figure 1 - Uni Pay class diagram.

This class diagram represents all the variables within the different methods that are present in the software. Each method carries a different task in which helps the program run smoother and provide a better experience for the user.

Sequence Diagram

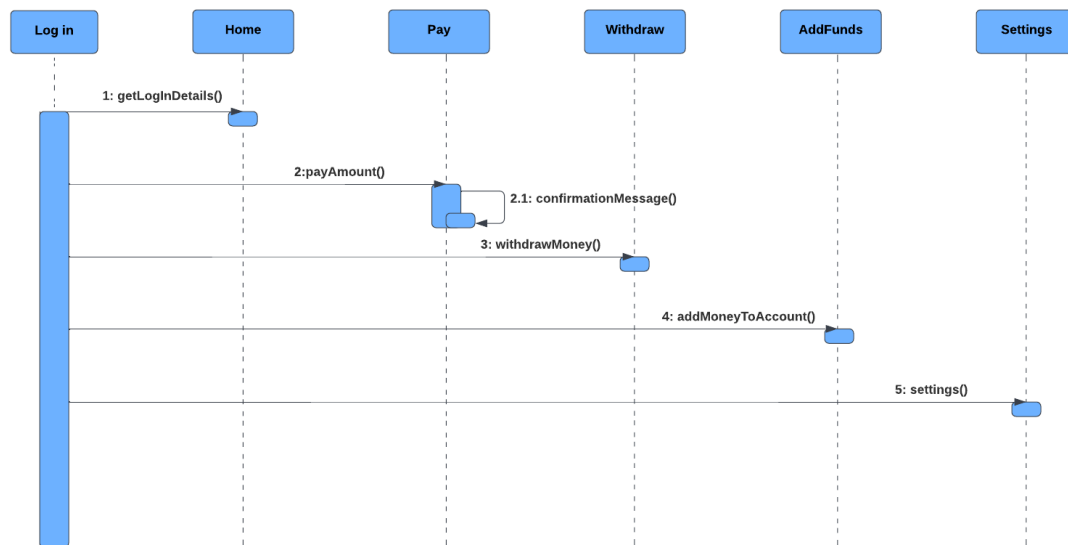
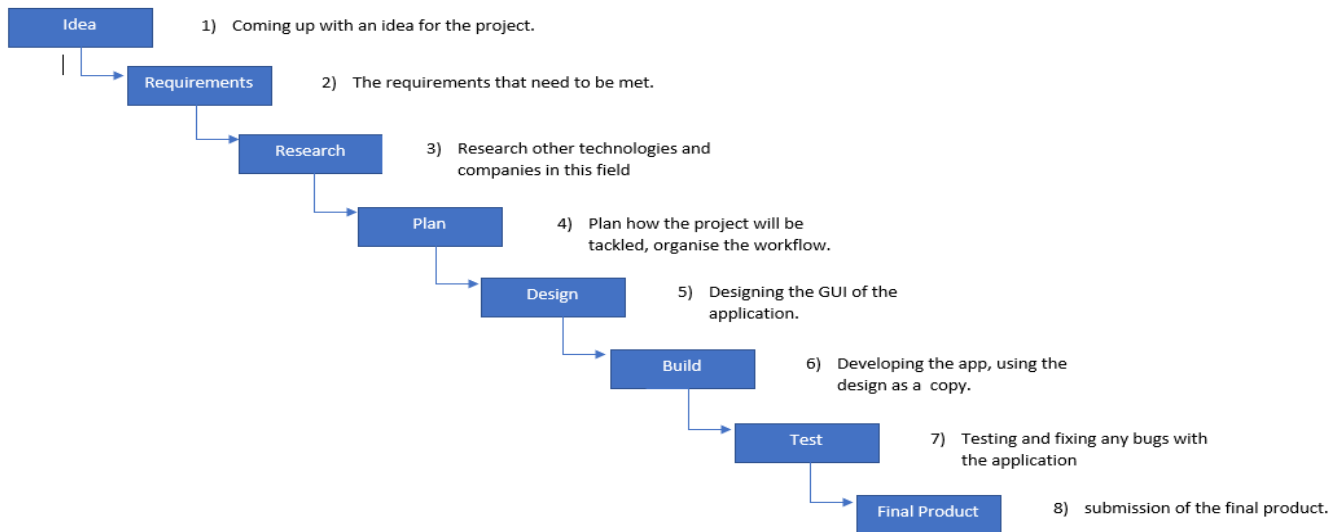


Figure 2 - Uni Pay sequence diagram

The sequence diagram reflects on the main functionalities of the software. It demonstrates how the user's actions operate in the program. After the user logs in, he will be taken to the home page which is also the main page of the software. If he wishes to make a payment, he will be taken to the payment page and if the transaction is successful, the user will receive a confirmation message and his payment page will be refreshed ready to make a new payment. And the other feature are very straight forward, where it takes the user onto the destined page in order to perform the task the user wants to carry.

Product management plan - Waterfall



Conclusion

In conclusion our team worked quite well together. We were able to distribute tasks efficiently and between the members we had all the skills necessary for the project. In terms of future development our team could improve teamwork and communication between the members. Due to some of the limitations during the development cycle of the product, some features were not able to be implemented, however the team will consider adding extra features in the future .Overall the project was a success and the team was able to develop a good product.

Bibliography

Robb, C. A. (2017). College student financial stress: Are the kids alright?. *Journal of Family and Economic Issues*, 38(4), 514-527.

Hejlsberg, A., Torgersen, M., Wiltamuth, S., & Golde, P. (2008). *The C# programming language*. Pearson Education.

Farrell, J. (2011). *Java programming*. Cengage Learning.

Microsoft. (2022). *Visual Studio*. <https://visualstudio.microsoft.com/vs/>

Thung, F., Bissyande, T. F., Lo, D., & Jiang, L. (2013, March). Network structure of social coding in github. In *2013 17th European conference on software maintenance and reengineering* (pp. 323-326). IEEE.

Appendices

Appendix A: Project Plan

Task To Complete	Deadline	Completed on time?
Think of an acceptable idea	Week 1	Yes
Research competitors	Week 2	Yes
Create Poster	Week 3	Yes
Present Poster	Week 4	Yes
Project Plan	Week 5	Yes
Group Report 1	Week 6	Yes
Peer Assess marking	Week 6	Yes
Program Development	Week 11	Yes
Program UI design	Week 11	Yes
Software Demo	Week 11	Yes
Group Report 2	Week 12	Yes

Appendix B: Agenda / Meeting Minutes

Meeting 4 Agenda:

Date: 02 March 2022

- Review assignment spec for the 2nd group report.
- Update progress on project plan.
- Discuss app creation

Meeting 4 Minutes:

Date: 02nd March 2022

Time: 4PM – 4:40PM

Attendance: Annum, Caleb, Nathan, Raul, Sam

1. Our group looked over the assignment rubrics and briefly discussed the content needed for the report.
2. We updated our project plan
 - We added the completion of group report 1.
 - We added the development of the app.
3. We discussed how our app would be created
 - Our app will be created using C#
 - Caleb took the lead on app development.

Meeting 5 Agenda:

Date: 16th March 2022

- Assign group tasks for the remainder of the term.
 - Group report 2 tasks
 - App design tasks
- Create a collaborative document to complete these tasks.

Meeting 5 Minutes:

Date: 16th March 22

Time: 4PM – 4:40PM

Attendance: Annum, Caleb, Nathan, Raul, Sam

1. We assigned individual tasks for group report 2
The tasks were added to the task assignment table.
2. Raul created a collaborative google document for both the group report and the task assignments so we could work individually on the same report.

Meeting 6 Agenda:

Date: 30th March 2022

- Test the software demo in Time for presentation.
- Discuss who is doing what during the app presentation.

Meeting 6 Minutes:

Date: 30th March 22

Time: 4PM – 4:40PM

Attendance: Annum, Caleb, Nathan, Raul, Sam

1. We tested the software demo to prepare for the presentation.
2. We then discussed how we would do the software demo and decided that Nathan would take the lead.
3. Nathan would also bring a laptop and cable in order to show the demo.

Meeting 7 Agenda:

Date: 6th April 2022

- Update on the progress of the group report.
- Update on the completion of the project plan.

Meeting 7 minutes:

Date: 6th April 2022

Time: 4PM – 4:40PM

Attendance: Annum, Caleb, Nathan, Raul, Sam

1. The group updated the project plan
 - Everything but the 2nd group report was completed.
2. The group reviewed the progress of the 2nd group report and confirmed that it would be completed in time.

Appendix C: Screenshots of Software

UniPay

Available Spend

£ 0

For Today

Current Balance

£ 0

Upcoming Payments

Add

Remove


Daily Spending


Budgeting Tips


1) Decide why your budgeting


2) prioritise expenses and goals


3) Revisit your Budget monthly

Pay

Withdraw

Home

Add Funds

Settings

UniPay

How much money do you earn a month ?
(Not including student loan)

£ 500

Student Loan income every 3 months
(keep 0 if not applicable)


£ 1200


Monthly Expences
eg gass, water, electric, rent (keep 0 if not applicable)


£ 80


Save Settings


(Version 1.0.0)

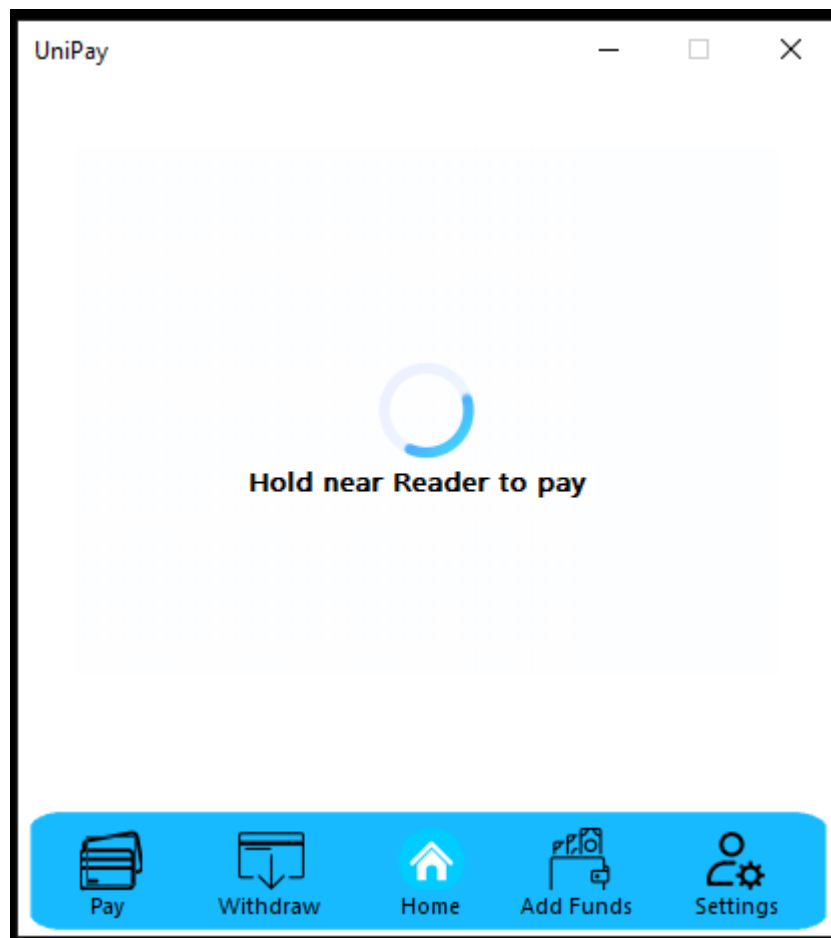
Pay

Withdraw

Home

Add Funds

Settings



UniPay

Available Spend	Current Balance
£ 0.4	£ 12
For Today	

£

Enter amount to add

First Name	Last Name
<input type="text" value="John"/>	<input type="text" value="Doe"/>

16 Digit Card number

Peer Assessment

Students	CB	RR	AM	SJ	NW	Total
Caleb	120	130	110	90	60	500
Raul	130	120	120	80	50	500
Annum	130	120	110	70	70	500
Samuel	130	120	110	70	70	500
Nathan	130	120	110	70	70	500
Total	640	610	560	370	320	2500