WALLE-TM

Bringing your E-wallet and ATM one step closer

VS Solutions

Project Report

MOTIVATION

This project mainly focuses on giving an alternative instead of solving a problem

As we all know that E-wallets are now one of the most preferable and trusted payment methods and a great way to carry money online and what if you can use your E-wallet for withdrawing money from ATM machine?

Yes, you heard right. We are trying to develop a new way to withdraw money from your ATM machine i.e. by using your E-wallets. This could revolutionize the dynamics between Digital money and physical cash.

This also addresses another problem i.e. cash withdrawal from our application doesn't require ATM card so even if you forget your ATM card at home it will not be a problem.

A well reputed bank and an E-wallet company have collaborated for this project and this is our task to build an application for them. Each transaction will cost 6% of the total amount withdrawn to user. The bank and the E-wallet company will meet their expenses and make profit from that revenue.

OBJECTIVE

The goal of this project is to do the project planning for developing an application which will act as an interface between your E-wallet and ATM and will make it possible to withdraw your digital money of E-wallet from ATM machine.

We have to do feasibility study for this project which is necessary as we should have an idea whether our project is economically and technically feasible or not.

We have to identify stakeholders and their responsibilities, activities and their duration, estimate the resources for the project.

We will do risk analysis and Cost benefit analysis to check whether the project is going to provide some profit or not

In the end we will evaluate our project with the help of some project evaluation techniques

The following milestones will be achieved during the project:

- Feasibility Study
- Project Plan
- Risk Identification
- Cost Benefit Analysis
- GANTT char and CPM/PERT for project evaluation

WALLE-TM

Bringing your E-wallet and ATM one step closer

TEAM MEMBERS

SHIVAM CHAUDHARY **2K17/SE/108**

VAIBHAV VERMA **2K17/SE/123**

SUBMITTED TO

Mrs. Juhi Jain

Software Project Management

Software Engineering (SE-A2)

Delhi Technological University

FEASIBILITY STUDY

The aim of feasibility study is to check whether the project is economically and technically feasible.

WHAT OUR APPLICATION MUST DO?

1. Support lakhs of transactions at the same time.

2. Find the nearest ATM for the transaction.

3. Provide an interface between the wallet and bank account of the user.

4. The U.I. should be easy to use since we are targeting a wide age group.

5.Security needs to be maintained to ensure the integrity of the data, although no malicious or advantageous editing is expected. Password-protection and a login system (based on access-level or user type) are sufficient.

6. Generate Robust OTP for the authentication process to help establish a connection between the bank and the ATM.

ECONOMIC FEASIBILITY

1. The initial investment will be done by the E-wallet company except for the manufacturing of QR codes for the ATM machine, which will be done by the bank.
2. The main source of revenue is the service charge of 6% on the total amount withdrawn by an individual.
3. Considering that we get 6% of the total amount withdrawn as revenue, initial investment can be easily recovered in just a few years.

4. So, we can say that our project is economically feasible.

TECHNICAL FEASIBILITY

1. For the front end we will use Angular9 framework.
2. For back-end spring will be used.
3. PostgreSQL will be used for maintaining the database.
4. AWS servers will be used.

So, to conclude we can say that our project is both economically and technically feasible.

PROJECT PLAN

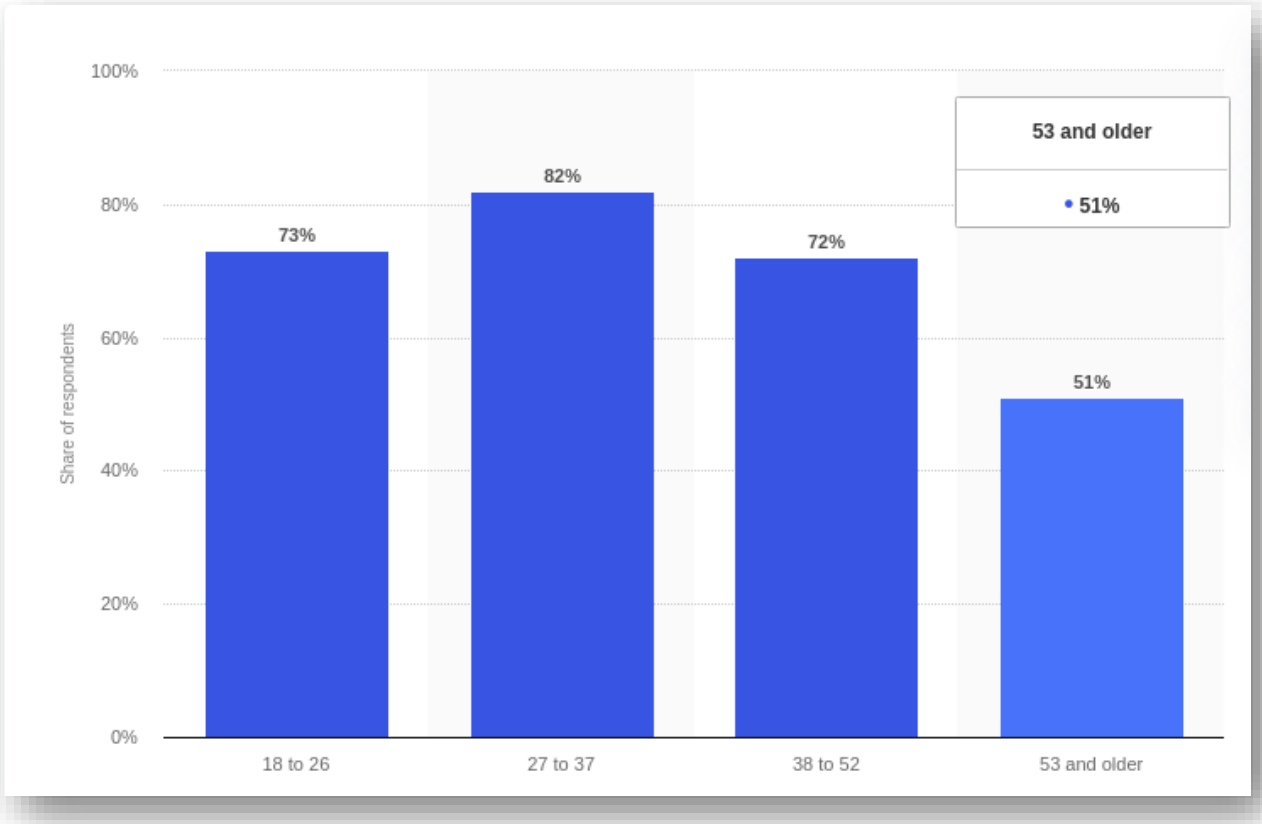
CURRENT MARKET SITUATION

Online payments have become a part of our daily lives. We're transacting online not just through debit or credit cards but there are numerous other modes like UPI, net-banking, and e-wallets as well.

The number of e-wallet users has increased dramatically over the years, these are some stats supporting our argument:

- Paytm has 100 Million (10 crores) installs on Android play store.
- Google pay has 100 Million+ installs on Android play store
- PhonePe also has 100 Million+ installs on Android play store.
- Mobiqwik also has 100 Million+ installs on Android play store.

.



Share of consumers using mobile wallets across India as of January 2018, by age group

This statistic depicts the results of a survey conducted in January 2018 about the share of consumers using mobile wallets across India, by age group. During the survey period, around 82 percent of banking consumers aged 27 to 37 stated that they used mobile wallets for payments across India.

So, introducing such a platform will be beneficial to all the e-wallet users and will set an example to motivate other e-wallet companies and banks to collaborate like this.

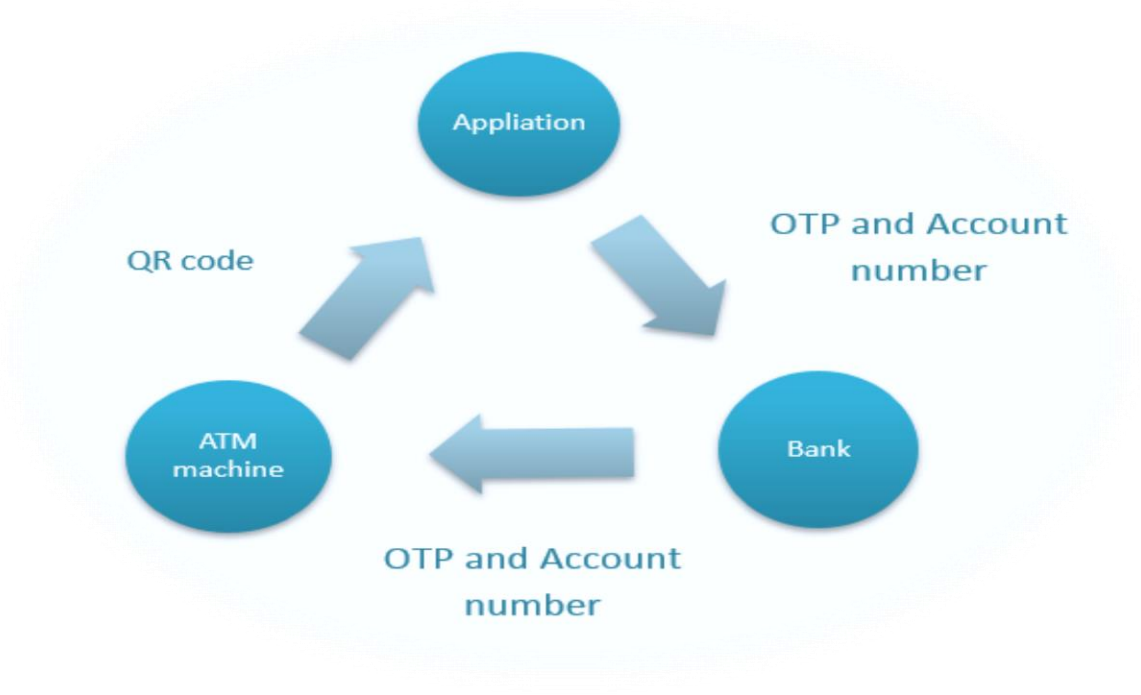
EXPECTED SOLUTION

All the e-wallets while doing KYC requests the details of the bank account of an individual. We can use that to our advantage. What people will feel is that they are able to transact the money directly from their e-wallets but the actual thing that will be happening is the money will be transferred to their account and then it is withdrawn from the ATM. This whole setup will behave like a black box where the input will be money from the E-wallet and output will be cash. There will be a unified service charge (5%) that will incorporate the cost bear by both the bank and the e-wallet company.

Procedure

- User should have account in the partner bank and E-wallet.
- All the ATM machines of the bank have a unique QR code attached to them which will contain information like the serial number of machine location of the machine etc.
- User will log in to the application for authentication.
- Now user can find a nearby ATM machine by the use of our application.
- User will scan the QR code attached to the machine. Now our application will know that the user wants to withdraw cash from that particular ATM machine.
- The application will generate an OTP and share it with bank and user.
- Along with OTP, user account number, serial number of ATM machine will also be shared with bank and bank in turn will share the OTP and user account number with the ATM machine.

- As soon as the user will enter the OTP in the ATM machine, they would be authenticated.
- Now, user will enter the amount to be withdrawn. A total of withdrawal amount+6% of withdrawal amount will be deducted from their e-wallet and transferred to their account.
- Bank will again communicate the amount with the ATM machine.
- Once the cash is ejected, user session will end.



This figure shows the flow of information between the ATM machine, Bank, Application

STAKEHOLDERS AND THEIR RESPONSIBILITIES

Bank

- Establish a fast and robust connection with WALLE-TM and ATM machines so that no error would occur during the transaction.
- Install QR codes for all the ATM machines which will determine each of the machines uniquely.
- Motivate more and more customers to use WALLE-TM.

E-wallet company

- Provide all the necessary information to the developers about their application so that the new Application can be developed.
- Motivate the customers without KYC to link their bank account with their E-wallet.
- Advertise about this new functionality.

Customers

- Give proper feedback to the concerned team for the betterment of the software.
- Do not try to misuse the machine i.e. not having the required amount and still trying to withdraw money. The software will handle this case anyway.

VS solutions:

- Build an efficient design for the Application.
- Generate robust O.T.P. such that no collision would occur.
- Establish a fast and strong connection between the bank and the e-wallet.
- The user interface should be easy to understand for every age group.
- Understand the requirements stated by both the bank and the e-wallet company properly.

IDENTIFICATION OF ACTIVITIES

- **Feasibility study:** Study the current market scenario and check whether the project is feasible or not.
- **Obtain requirements:** gather technical and operational requirements from both the parties i.e. the bank and the e-wallet company.
- **Product analysis:** Complete analysis of the application model. We will be analyzing all the techniques and possible methods to find the one that meets the required security and performance measures.
- **Product design:** We will identify the different modules to be created, how they will interact, the level of cohesion and coupling between the modules, the total number of U.I. screens, etc.
- **Develop database:** Create a database mapping the user's bank account to the user's e-wallet and other important information.
- **ATM Upgradation:** In the meantime, the bank will arrange the QR codes for all the ATM machines containing useful information about the individual machines. Also, a few changes in the ATM machine User Interface will be done to incorporate the OTP screen.
- **Implementation:** After the product analysis and product design is done, we will move towards the implementation phase.
- **Testing:** There is a lot to test i.e. the QR codes the communication between e-wallet, our app, bank, etc. and the unit testing for the individual modules of the application.
- **Advertisement:** Advertisement before the product is launched so that when it is launched people already know about it.
- **Deployment**: Finally deploy the application for the final use.

DURATION OF ACTIVITIES

Activity	Description	Required Predecessor	Duration (weeks)
A	Feasibility study	-	**4**
B	Obtain requirements	-	6
C	Product analysis:	B	2
D	Product design:	C	4
E	Develop database	B	2
F	ATM Upgradation	A	8
G	Implementation	D, E	10
H	Testing	G, F	4
I	Advertisement	H	2
J	Deployment	I	2

ESTIMATION OF RESOURCES

FUNCTION POINT ANALYSIS:

Functional Unit	Count	Complexity	Complexity total	Functional Unit Total
External Inputs (EI)	1	Low * 3	3	17
	2	Average * 4	8	
	1	High * 6	6	
External Outputs (EO)	1	Low * 4	4	21
	2	Average * 5	10	
	1	High * 7	7	
External Inquiries (EI)	1	Low * 3	3	9
	0	Average * 4	0	
	1	High * 6	6	
Internal Logical Files (ILF)	1	Low * 7	7	17
	1	Average * 10	10	
	0	High * 15	0	
External Interface Files (EIF)	2	Low * 5	10	20
	0	Average * 7	0	
	1	High * 10	10	

ILF = 17

EIF = 21

External Input = 9

External Output = 17

User Enquiries = 20

UFP = 84

CAF = 0.65 + 0.54 = 1.19

FPA = 84*1.19

FPA = 100

Using the total number of Functional Points and considering our main language as JAVA the **SLOC** calculated is**: 5KLOC**

SEL MODEL:

$E=1.4L^{0.93}$

$DOC=30.4L^{0.90}$

$D=4.6L^{0.26}$

E= Efforts (Person-Month)
DOC=Documentation (Number of Pages)
D = Duration (D, in months)
L = Number of Lines per code in KLOC

For our project:

L = 5 KLOC.

Effort = 6.25 Person-Month.

DOC = 129.4 Number of pages

Duration = 6.99 months

Persons Required = 1 Person

Productivity = 0.8 KLOC/Person-month

WALSTON FELIX MODEL

WALSTON and FELIX develop the models at IBM provide the following equation gives a relationship between lines of source code and effort:

$$E=5.2L^{0.91}$$

In the same manner duration of development is given by

$$D=4.1L^{0.36}$$

For our project:

Effort = 22.5 Person-Months.

Productivity = 0.22 KLOC/Person-Months

Person Required = 3 Persons

Duration = 7.32 months.

BASIC COCOMO

Effort = $a*(KLOC)^b$

Time = $c*(Effort)^d$

Person Required = Effort / Time

Productivity = KLOC / Effort

The above formula is used for the cost estimation of the basic COCOMO model, and also is used in the subsequent models. The constant values a, b, c and d for the Basic Model for the different categories of the system:

A = 3 B = 1.12 C = 2.5 D = 0.35

SOFTWARE PROJECTS	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

EFFORT: 13 Person - months

Duration: 6.62 Months

Person Required: 13/6.62 = 1.96 Persons ~ 2 Persons

Productivity: 5/13 = 0.38 KLOC / Person-Months

INTERMEDIATE COCOMO

Formulas use:

Effort = $a*(KLOC)^b * EAF$

Time = $c*(Effort)^d$

Person Required = Effort / Time

Productivity = KLOC / Effort

EAF: Effort Estimation Factor

The basic Cocomo model considers that the effort is only a function of the number of lines of code and some constants calculated according to the various software systems. The intermediate COCOMO model recognizes these facts and refines the initial estimates obtained through the basic COCOMO model by using a set of 15 cost drivers based on various attributes of software engineering.

Classification of Cost Drivers and their attributes:

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

1.Product attributes -

Required software reliability extent: **1.40 (Very High)**

Size of the application database: **1.08 (high)**

The complexity of the product: **1.00 (Nominal)**

2.Hardware attributes -

Run-time performance constraints: **1.11 (High)**

Memory constraints: **1 (nominal)**

The volatility of the virtual machine environment **1(Nominal)**

Required turnabout time: **1.07 (High)**

3.Personnel attributes -

Analyst capability: **1 (nominal)**

Software engineering capability: **0.86 (high)**

Applications experience: **0.91 (High)**

Virtual machine experience: **1 (Nominal)**

Programming language experience: **0.95 (high)**

4.Project attributes -

Use of software tools: **1 (Nominal)**

Application of software engineering methods: **0.91 (High)**

Required development: **1(Nominal)**

EAF: **1.214**

Effort = 13 X EAF = **15.78 person-months**

Time = c*(Effort)^{d} = **7.13 months.**

Person required: **2 Persons**

Productivity = **0.31 kloc/person-months**

DETAILED COCOMO

The basis resources will be same as intermediate Cocomo model

EAF: 1.214

Effort = 13 X EAF = 15.78 person-months

Time = c*(Effort)^{d} = 7.13 months.

Person required: 2 Persons

Productivity = 0.31 **KLOC / Person-Months.**

Lifecycle Phase Values of μ_p

Mode & Code Size	Plan & Requirements	System Design	Detailed Design	Module Code & Test	Integration & Test
Organic Small S≈2	0.06	0.16	0.26	0.42	0.16
Organic medium S≈32	0.06	0.16	0.24	0.38	0.22
Semidetached medium S≈32	0.07	0.17	0.25	0.33	0.25
Semidetached large S≈128	0.07	0.17	0.24	0.31	0.28
Embedded large S≈128	0.08	0.18	0.25	0.26	0.31
Embedded extra large S≈320	0.08	0.18	0.24	0.24	0.34

Phase-wise effort:

- Planning and Requirements: 0.94 person-months
- System Design: 2.52 person-months
- Detail Design: 4.10 person-months
- Module Code and Test: 6.62 person-months
- Integration and Test: 2.50 person-months

Lifecycle Phase Values of τ_p

Mode & Code Size	Plan & Requirements	System Design	Detailed Design	Module Code & Test	Integration & Test
Organic Small S≈2	0.10	0.19	0.24	0.39	0.18
Organic medium S≈32	0.12	0.19	0.21	0.34	0.26
Semidetached medium S≈32	0.20	0.26	0.21	0.27	0.26
Semidetached large S≈128	0.22	0.27	0.19	0.25	0.29
Embedded large S≈128	0.36	0.36	0.18	0.18	0.28
Embedded extra large S≈320	0.40	0.38	0.16	0.16	0.30

Phase-wise development time:

- Planning and Requirements: 0.7 Months
- System Design: 1.3 Months
- Detail Design: 1.71 Months
- Module Code and Test: 2.78 Months
- Integration and Test: 1.28 Months

COCOMO II

Early Design model

PM nominal = $A*(size)^B$

PM nominal = Effort of the project in person months

A = Constant representing the nominal productivity, provisionally set to 2.5

B = Scale factor

Size = Software size

Precedent ness - very high (1.24)
Development Flexibility - very high (1.01)
Architecture/ Risk resolution – high (2.83)
Team cohesion - very high (1.10)
Process maturity - very high (1.56)

B = 0.91 + 0.01* 7.29 = 0.98

PM nominal = $2.5*5^{0.98}$ = 12.10 person-month

PM adjusted = PM nominal × EM

Early design Cost drivers	*Extra Low*	*Very Low*	*Low*	*Nominal*	*High*	*Very High*	*Extra High*
RCPX	.73	.81	.98	1.0	1.30	1.74	2.38
RUSE	—	—	0.95	1.0	1.07	1.15	1.24
PDIF	—	—	0.87	1.0	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.0	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.0	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.0	0.87	0.73	0.62
SCED	—	1.43	1.14	1.0	1.0	1.0	—

RCPX: nominal = 1
RUSE: low = 0.95
PDIF: high = 1.29
PERS: high = 0.83
PREX: high = 0.87
FCIL: nominal = 1
SCED: low = 1.14

Effort multiplier = **1.01**
PM adjusted = PM nominal * Effort multiplier = 12.10*1.01 = 12.221
PM adjusted = **12.221 person-months.**

OBJECTED ORIENTED ESTIMATION

Use Case Point Method:

Actors	Complexity	Weight
Bank	Simple	1
ATM	Complex	3
Customer	Average	2
Admin	Average	2

Unadjusted Actor Weight = 8

Use case weight

2 X 5 = 10

7 X 10 = 70

6 X 15 = 90

UUCW = 170

UUCP = 178

TCF = 0.6 + 0.49 = 1.09

ECF = 1.13

UCP = 178 X 1.09 X 1.13 = 219.24

Effort = 4384 Person Hour

RISK ANALYSIS

RISK IDENTIFICATION

1. Coding takes longer than expected.
2. Team members get ill (Covid infected).
3. Design is not robust.
4. Documentation is improper.
5. Requirement Specification takes longer than expected.
6. Testing reveals a lot of bugs.
7. Lack of experience causes delays in projects.
8. User interface might be complicated for some age group
9. Real-time performance shortfall: More time consumed than expected when operations performed on the field making the process inconvenient.
10. Technologies used might get outdated.
11. Clashes in ideologies of Bank and e-wallet company.
12. If users are not satisfied with the idea then the company might face a huge loss.
13. Security should be robust, otherwise, the system gets susceptible to hackers.
14. App crashes due to heavy load (Scalability issues).

RISK PRIORITIZATION

S. No	Problem	Probability of occurrence	Impact	Risk exposure	Priority
R1	Coding takes longer than expected.	4	6	24	2
R2	Team members get ill (Covid infected).	6	5	30	1
R3	Design is not robust.	2	9	18	4
R4	Documentation is improper.	2	4	8	11
R5	Change in Requirement during implementation	4	4	16	5
R6	Testing reveals a lot of bugs.	1	9	9	10
R7	Lack of experience causes delays in projects.	2	10	20	3
R8	The user interface might be complicated for some age group	3	4	12	8
R9	Real time performance shortfall	1	7	7	12
R10	Technologies used might get outdated.	3	5	15	6
R11	Clashes in ideologies of Bank and e-wallet company.	2	3	6	13
R12	Security should be robust	1	10	10	9
R13	Another competitor might appear.	2	7	14	7
R14	App crashes due to heavy load (Scalability issues).	1	5	5	14

RISK MATRIX

Urgent (5-10)	**R3, R6, R7, R12, R9, R13, R14**	**R1, R10**	**R2**	
High (3-4)	**R4, R11**	**R5, R8**		
Medium (1-2)				
Low (0)				
	Low (0-2)	Medium (3-5)	High (6-7)	Urgent (8-10)

Impact of risk

Probability of occurrence

RISK MITIGATION TECHNIQUES

S.No	Problem	Risk mitigation techniques
R1	Coding takes longer than expected.	1. Maintaining checkpoints with projected time for each section of code. 2. Motivating the developers to give their 100%
R2	Team members get ill (Covid infected).	1. Regular health check-ups for staff. 2. Maintaining proper hygiene at the workplace. 3. More Emphasis on work from home
R3	Design is not robust.	1. Do all the design related checks prior to finalizing the design 2. Proper verification of Deliverables with performance estimation
R6	Testing reveals a lot of bugs.	1. After developing each unit do a little unit testing to ensure the unit doesn't contain too many bugs.
R7	Lack of experience causes delays in projects.	1. Make sure that the staff is well experienced. 2. Proper training of Staff on technologies used.
R9	Real time performance shortfall	1. Run several kinds of simulations before deploying the product. 2. Do not make unattainable claims about the performance.
R10	Technologies used might get outdated.	1. Do a proper brainstorming session with all the team members about which technologies to use.
R12	Security should be robust, otherwise, the system gets susceptible to hackers.	1. Hire ethical hackers for testing system security.
R13	Another competitor might appear.	1. Extensive research and surveys in order to get a upper hand on the domain.
R14	App crashes due to heavy load (Scalability issues).	1. Make the system highly scalable as, the number of transactions involved will be very large.

MONITORING THE PROJECT

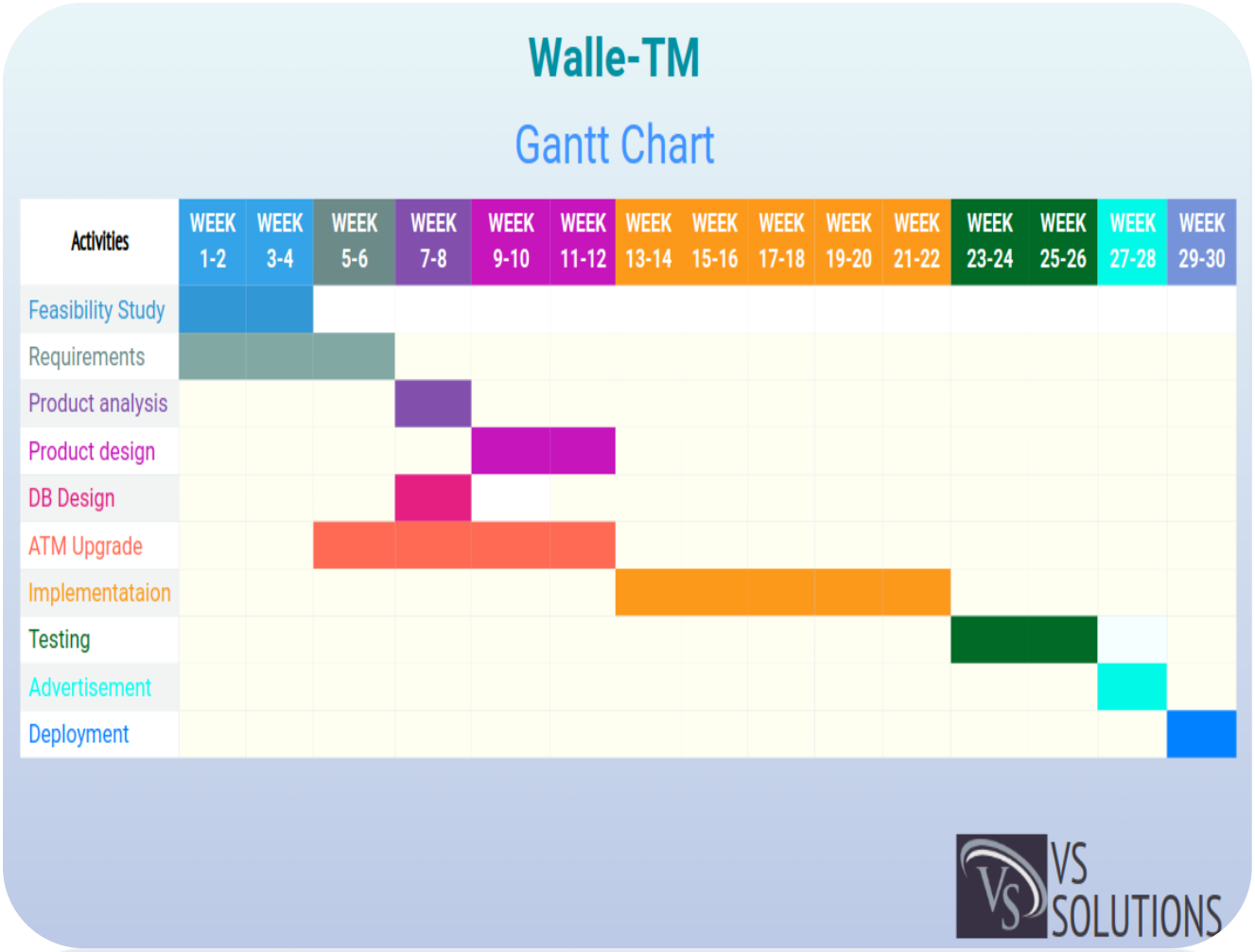
GANTT CHART

Schedule of activities:

Activity	Description	Required Predecessor	Duration (months)
A	4	-	2
B	6	-	3
C	2	B	1
D	4	C	2
E	2	B	1
F	8	A	4
G	10	D, E	5
H	4	G, F	2
I	2	H	1
J	2	I	1

We can make GANTT chart by referring to this table

GANTT CHART


Walle-TM
Gantt Chart
Activities
WEEK 1-2
WEEK 3-4
WEEK 5-6
WEEK 7-8
WEEK 9-10
WEEK 11-12
WEEK 13-14
WEEK 15-16
WEEK 17-18
WEEK 19-20
WEEK 21-22
WEEK 23-24
WEEK 25-26
WEEK 27-28
WEEK 29-30
Feasibility Study
Requirements
Product analysis
Product design
DB Design
ATM Upgrade
Implementataion
Testing
Advertisement
Deployment
VS SOLUTIONS

CRITICAL PATH METHOD (CPM) CHART

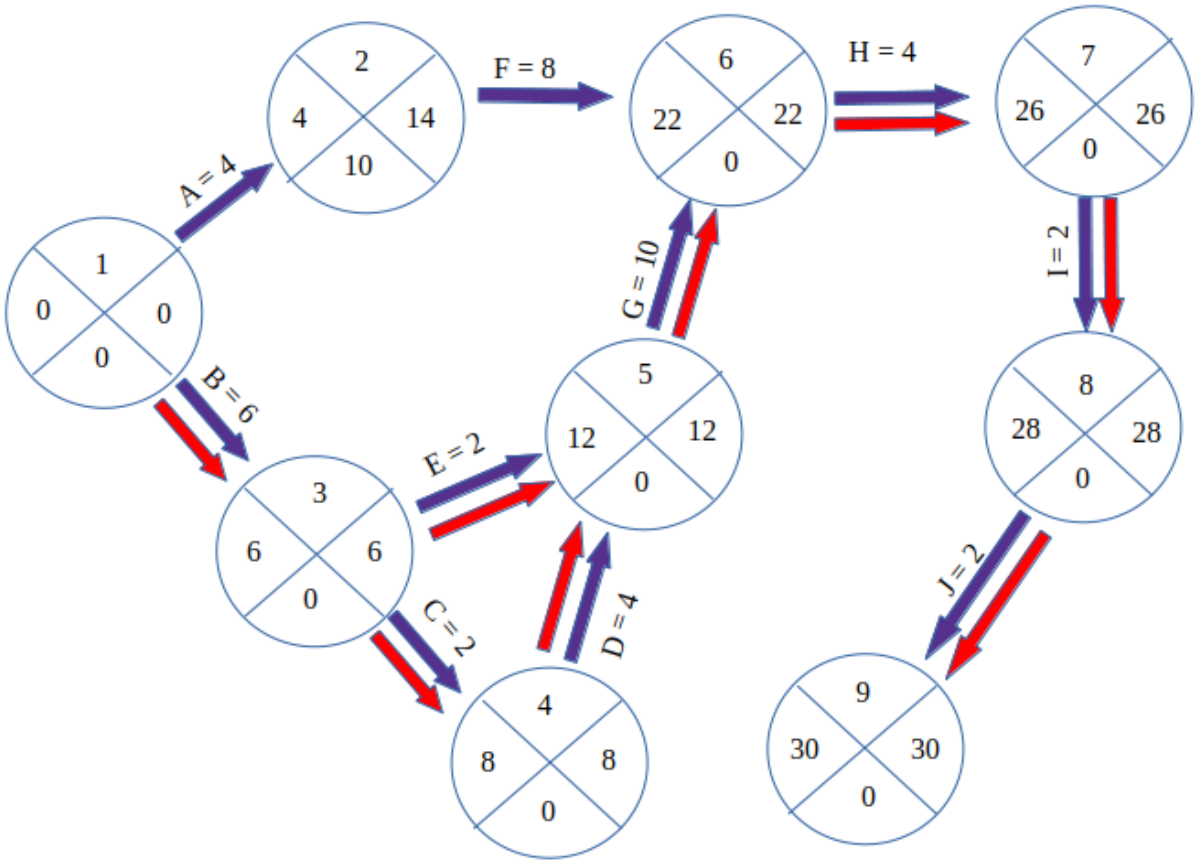
Table After Forward Pass

Activity	Duration (Weeks)	Required Predecessor	Earliest Start date	Earliest End date
A	4	-	0	4
B	6	-	0	6
C	2	B	6	8
D	4	C	8	12
E	2	B	6	8
F	8	A	4	12
G	10	D, E	12	22
H	4	G, F	22	26
I	2	H	26	28
J	2	I	28	30

Table After Backward Pass

Activity	Duration (Weeks)	Earliest Start date	Earliest End date	Latest Start Date	Latest End Date
A	4	0	4	0	4
B	6	0	6	0	6
C	2	6	8	6	8
D	4	8	12	8	12
E	2	6	8	10	12
F	8	4	12	14	22
G	10	12	22	12	22
H	4	22	26	22	26
I	2	26	28	26	28
J	2	28	30	28	30

CPM CHART



As we can see there are two critical paths:

- 1 – 3 – 4 – 5 – 6 – 7 – 8 – 9
- 1– 3 – 5 – 6 – 7 – 8 – 9

PERT CHART

$T = (a + 4m + b)/6$

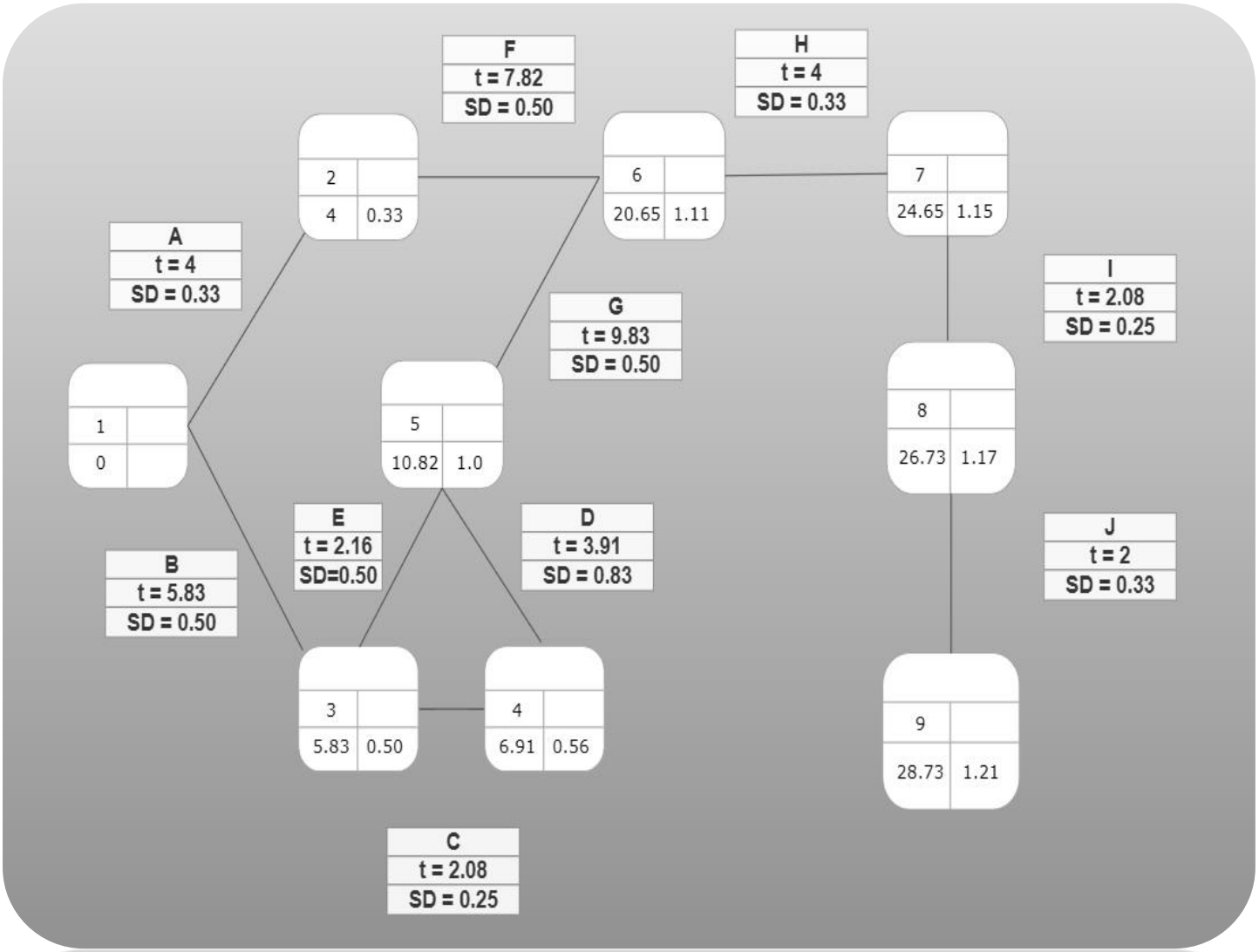
$SD = (b-a)/6$

Where

T: total estimated time

SD: Standard deviation

Activity	Description	Optimistic (a)	Most Likely (m)	Pessimistic (b)	T	Standard Deviation (SD)
A	Feasibility study	3	4	5	4	0.33
B	Obtain requirements	4	6	7	5.83	0.50
C	Product analysis:	1.5	2	3	2.08	0.25
D	Product design:	2.5	4	5	3.91	0.83
E	Develop database	1	2	4	2.16	0.50
F	ATM upgradation	6	8	9	7.83	0.50
G	Implementation	8	10	11	9.83	0.50
H	Testing	3	4	5	4	0.33
I	Advertisement	1.5	2	3	2.08	0.25
J	Deployment	1	2	3	2	0.33



Pert chart

TT = 30 weeks
Z = (TT – t)/ SD
Z = 1.05.
Probability of completion of project on time = **85.31%**
Risk = **14.59 %**

TT = 31 weeks
Z = (TT – t)/ SD
Z = 1.88.
Probability of completion of project on time = **97%**
Risk = **3%**

COST BENEFIT ANALYSIS

ASSUMPTIONS

1. We are assuming that as of now there are 2 lakh customers which use both the bank and the e-wallet and have done their KYC.

2. This is the assumed % of users using this feature in all 5 years:
   Year1 - 20%
   Year2 - 30%
   Year3 - 28%
   Year4 - 35%
   Year5 - 42%

3. This is the expenditure for maintenance assumed for next 5 years:
   Year1 - 50,000
   Year2 - 60,000
   Year3 - 50,000
   Year4 - 40,000
   Year5 - 50,000

4. We are assuming that on average, a customer will withdraw an amount of Rs 100. So, for next 5 years total profit will be:
   Year1 – 2,80,000 - 50,000 = 2,30,000
   Year2 – 4,20,000 - 60,000 = 3,60,000
   Year3 – 3,92,000 - 50,000 = 3,42,000
   Year4 – 4,90,000 - 40,000 = 4,50,000
   Year5 – 5,88,000 - 50,000 = 5,38,000

5. We have assumed the Rate of interest to be 8%

NET PRESENT VALUE

Year	Cash Flow	(Discount Factor: 8%)	Discounted Cash Flow
0	-7,00,000	1.00	-7,00,000
1	2,30,000	0.9259	2,12,957
2	3,60,000	0.8573	3,08,628
3	3,42,000	0.7938	2,71,480
4	4,50,000	0.7350	3,30,750
5	5,38,000	0.6806	3,66,163
Net Profit	12,20,000	NPV	7,89,978

FINDINGS

1. Net Profit**:** sum of cash flow over all 5 years = **12,20,000.**

2. RETURN on INVESTMENT:

   ROI = Average Annual Profit / Total Investment * 100 = **34%**

3. Net Present Value: Sum of all discounted cash flows = **7,89,978.**

CONCLUSION AND FUTURE EXPANSION

This is a very new project in its domain, to predict its outcome will be very difficult but, one thing we can say for sure is that this will inspire a lot of entrepreneurs to take the risk and try their hand in this field.

If everything works out fine (since we have done basic risk analysis, cost benefit analysis) then, this project will be a huge success and there are a lot of things that we can try which are:

- Not mandating the customer to have account in the partner bank in order to withdraw money.
- Making a unified E-wallet which can interact with any bank's ATM machine.
- Instead of making it a two-way process (amount first goes into user's bank account and then it is withdrawn) we can directly use the amount in E-wallet to withdraw the money.

Nobody has tried their hands in this area so there is a huge chance of expansion and we can always try new things.

INDIVIDUAL CONTRIBUTION

This whole project was done on voice chat on discord and google sheets so specifying individual contribution is kind of impossible.

In each task from the starting till the end of the project both of us were involved so, to say that this particular task is done by one of us would be an insult to another.

REFERENCES

smartdraw.com – for making PERT chart

Vengage.com – for making GANTT chart

APPENDIX A(PLAGIARISM REPORT)