Access To Fund (A2F)

A COMMUNITY BASED CROWDFUNDING PLATFORM

SE 505 : Software Project Lab II

Submitted by

Khondaker Tasnia Hoque

BSSE Roll No. : 1205

Sifat Sikder

BSSE Roll No. : 1221

Supervised by

Dr. Kazi Muheymin-Us-Sakib

Designation: Professor

Institute of Information Technology

Institute of Information Technology

University of Dhaka

29-05-2023

Supervisor's Signature: _____

Access to Funds (A2F)

---

1. Introduction

This chapter is a part of our software requirement specification for the project "A2F". In this chapter, we have focused on the intended audience for our project "A2F"

1.1 Purpose

This document describes the Software Requirement Analysis of A2F application. It contains all the functional, non-functional, supporting requirements and expected requirements. At the same time it establishes a requirement's baseline for the development of the project. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. But as time goes on, our SRS document is expected to evolve as users and developers work together to validate, clarify and expand its contents.The SRS serves as an official medium of expressing users' requirements to the developer and provides a common reference point for both the developer team and the stakeholder community.

1.2 Intended Audience

This SRS report is intended for several audiences which includes the end users as well as the project managers, designers, developers, and testers.

- With the use of this A2F SRS document, the developer team's work implies that the customer can simply determine whether the product they have developed is acceptable to them or not.

- Project Managers will be able to plan milestones, delivery date and at the same time ensure whether the developing team is on track or not during the development of the application by using this SRS document.
- The designers will use this SRS as a basis for creating the application's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will make use of this SRS document to create the system's functionality. To ensure that the software they develop will satisfy all of the customers' documented criteria, the developers will link the requirements outlined in this SRS to the product they construct.
- The testers will use this SRS during various testing rounds. For each requirement that has been established, they will use this SRS to generate test plans and test cases. The testers will conduct their tests on the software after portions of it are finished to make sure it satisfies the specifications mentioned in this SRS. When the system is finished, the testers will rerun their tests to make sure that all the requirements outlined in this SRS have been met.

1.3 Conclusion

The analysis helped us figure out the stakeholders and requirements systematically. Moreover, it helped us to focus on the users who will be using our analysis. The document will be helpful to each and every stakeholder of the software to understand, relate and ensure convenient usage of the software.Developers and testers will have some clear ideas of what to do and development will be smoother. More and more communication between the stakeholders and developers will help evolve the software and enhance it further.

2. Inception

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At

project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.
To establish the groundwork, the following factors have been worked on to the inception phase:

- Icebreaking
- List of stakeholders.
- Recognizing multiple viewpoints.
- Working towards collaboration.
- Requirements questionnaire.

2.1 Icebreaking

The term "icebreaking" describes the process of lowering the communication gap between you and the other person. It is an important component because it indicates that our proposal has been accepted. We began by conversing with them in languages devoid of context. This stage is the only factor that determines how they act, react to our questions, or are open to making changes in their shops.

2.2 List of Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly.Stakeholders include both the system's end-users and everyone else in the organization who might be impacted by its installation. A list of those who will provide input is compiled at the beginning as needed.The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with some teachers and students and found some requirements.

We identified the following stakeholders for our “A2F” project.

- Teacher
- Student

2.3 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

Student’s viewpoint: (Borrower perspective)

- The app should give a small amount of loan anytime they demand.
- Loan taking process should be easy.
- Loan taking process should be less time consuming.
- Minimum maintenance fee.
- If any student suffers from disease or any unwanted situation, the app should arrange for fundraising.
- The App should give an opportunity to invest money so that students can get profit.

2.4 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting
requirements of our stakeholders. That’s why we followed the following steps to merge these
requirements-

- Find the common and conflicting requirements.
- Categorize them.
- List the requirements based on stakeholder’s priority points.
- Make final decisions about requirements.

Common viewpoints

- Easy to use UI
- Error free detection
- Verifying the suspicious activity

Conflicting viewpoints

There are no conflicting viewpoints among our stakeholders.

Final requirements

We finalize the following requirements based on stakeholder's priority point:

- User friendly system
- Authentication
- Error free system (maximum 10% error may be considerable)
- Easy to maintain software
- Keep records of all transactions.

2.5 Conclusion

The Inception phase helped us to establish a basic understanding about the loan management , identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

3.Elicitation

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

3.1 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.1.1 Collaborative Requirements Gathering

We have met with some stakeholders in the Inception phase such as the teachers and students. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are acting a vital role in the whole process) again to
elicit the requirements.

3.1.2 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the clients into technical requirements for the software. The prime concern of the QFD is customer satisfaction maximization. In order

to ensure this, QFD enforces an understanding of what customers describe as 'valuable' and then deploy these values
throughout the engineering process.

QFD defines three types of requirements:

- Normal Requirements
- Expected Requirements
- Exciting Requirements

Normal Requirements : Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for our project.

1. Authentication
2. Cash In
3. Investment
4. Loan request posting
5. Repayment deadline notification
6. Setting penalty if loan not be able to repay in due time

Expected Requirement : The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite not being explicitly mentioned, their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below:

1. Using google oauth for sign up
2. Store hashed password
3. Interactive and user friendly interface
4. 2 Factor authentication via NID registration
5. User rating

Exciting Requirement : These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project.

1. Fundraising for medical purposes
2. User Transactional statistics—---expected
3. System monthly transactional statistics—--expected
4. Making wallet to wallet transaction
5. Loan Prediction for user

Overview (user perspective)

Depositor	Borrower
Cash In	Giving Post for loan
Invest money in system	Select instant loan option
Bidding	Select bidding loan option
Donation	Return loan

Overview (system perspective)

Send money to borrowers account

Give rating to user

Send notification of borrower's post to depositors

Send warning to borrowers before 3 days of deadline

Set penalty to borrowers once the deadline is missed

Give Predictions to users

Make monthly transactional statistics

Block Diagram of A2F

4. Usage Scenario

A2F Introduction

A2F is a community based lending application using crowdfunding concept. It is a mobile application where funds will be available anywhere and anytime. We have recently performed a survey regarding short-term loans between IIT running batch students.We discovered that 68% of students take short-term loans for mostly academic and medical purposes. They suffer from availability of funds when it is really needed.

Also,A2F will raise funds in any kind of emergency situation(medical purposes). Our application A2F aims to solve this problem for accessing funds. Thus the mission of A2F is to provide short term loans within a specified community and also raise funds for that community welfare.

1. Features : The whole application is developed in 6 phases namely -

- Account Management
- Money Acquisition
- Transaction management
- Loan Management
- Loan repayment
- Profit distribution

Phase 1-Account Management

In this section we will describe different types of users in A2F and the user authentication process.There will be four types of users:

- Admin
- Agent
- Committee member

- Normal user

A user will enter the application either by sign up or login.

Sign up : All the users have to provide the following information to sign up to the system:

- Name
- Institutional Email/University Registration ID
- Phone No
- Password
- NID Card (Photo)

Login : All the users can log in with their email and password.

Password Recovery : If any user forgets his password, he can recover his passwords by email verification.

User Type Change: Initially every user will be a normal user.Only Admin can assign a normal user as committee member,admin,agent.New admins and agents will be selected from committee members by admins.

Phase 2 -Transaction Management

In this section we will look at the protocols of cash flows in and out of the app.We will also describe the A2F wallet types and its parts.

Cash Flow Protocols: There will be 2 types of transactions in A2F namely-

- **Physical transaction:** Any kind of transaction outside the application will be known as physical cash flow.
- **Logical transaction:** Any kind of transaction inside the application will be known as logical cash flow.

Wallet Description: There will be 2 types of wallet in A2F. One is a user wallet and another one is an app wallet.There will be 2 functions of a wallet namely-CashIn,CashOut. All 4 parts are described below:

CashIn: Cash in protocols are described below-

1. A user will do a physical transaction where he will send money from bKash wallet to A2F wallet using A2F's bKash web hook.
2. After a successful physical transaction, a logical transaction will happen. In the logical transaction, the 'cash in' amount will be added in the user wallet.

CashOut: Cash out protocols are described below-

1. User will press the cashout button and enter the amount he wants to cash out
2. A notification will be sent to the agent to complete the physical transaction.
3. The agent will send the money to the user's bKash account.
4. After a successful physical transaction the agent will mark the notification as completed.
5. When the agent marks the transaction as completed a logical transaction will happen where the money will be deducted from the user wallet.
6. The transaction will be stored in the user transaction history as well as in the app transaction history.

Phase 3-Money Acquisition

In this phase we will talk about how A2F will collect money from our committee members as well as from other investors. A2F wallet will mainly acquire funds from 2 processes namely-Call for Money and Normal Investment.These 2 processes are described below-

Call for Money

A2F will collect most of its needed funds by doing a call for money.The funds collected from the call for money will be given back to the investors at year end with profits. The steps of call for money are mentioned below:-

1. Admins will create a circular which will contain information about needed amount,starting and ending date of participation in call for money.
2. After creation,the circular will be shown to the dashboards of every user.
3. After that, users can invest money in response to that circular.In order to do so, a user will click on that circular. Then A2F will ask him to enter the amount he wants to invest.
4. After entering the amount,a logical transaction will happen from that user wallet to the app wallet which means the invested amount will be deducted from the user wallet and added to the app wallet and that amount will be shown in his investment dashboard.
5. After each successful transaction,the needed amount will be deducted from the circular.
6. The circular will be deleted from the dashboards when the needed amount is fulfilled.

Investment for normal users

1. Any user who wants to invest his fund will simply go to the invest interface.
2. There he will enter the amount and the time period of his investment and then press on the invest button.
3. After that, the user will get a notification about his investment.

Phase 4-Loan Distribution

In this phase we will demonstrate how A2F provides loans to its borrowers.Steps of this phase are mentioned below-

1.When a user wants to take a loan, he will go to the borrow interface.

2.Then he will fill up a loan application form. The loan application form will contain information about loan amount and reason why he needs the money.

The user will also define if he wants to repay the loan in installments or in bullet repayment(repay full amount after 6 months).

3.After providing all information the borrower will get a loan submission status(initially pending).

4.Based on the asking amount the loan request can be classified into 2 types namely-

Loan amount less than threshold

i.If the asking loan amount is less than the threshold amount(maximum amount to get loan directly from A2F) , the loan request will be accepted immediately.

ii. After acceptance, a logical transaction will take place from app wallet to borrower wallet.

iii. At the same time,our agent will get a notification that a loan request has been accepted. Then he will transfer the physical money from the app's bKash account to the borrower's bKash account.

Loan amount greater than threshold

i.If the asking amount is greater than the threshold amount, a loan post will be shown in the dashboard of all users.

ii. This loan post will be activated in their dashboards for 7 days.

iii. The committee members as well as other users can invest money in that loan post.

iv. All invested money will be stored in A2F's app wallet.

v.If the required fund is raised by all users within 7 days, the borrower will get a notification about loan request acceptance.Then the asking amount will be added to his wallet and deducted from the app wallet.

vi. At the same time,our agent will get a notification that a loan request has been accepted. Then he will transfer the physical money from the app's bKash account to the borrower's bKash account.

vii. As soon as the required loan amount is fulfilled, the loan post will be deleted from the dashboard of all users.

viii. On the other hand, if the fund is not raised within 7 days the loan post will be deleted from the dashboard and the borrower will get a notification about loan rejection.

Phase 5-Loan Repayment

In this phase, we will talk about how the borrower will repay the loan amount.If a user repays his loan amount within 1 month after loan taking,he will only have to repay the principal amount without any interest.Otherwise,the user can repay his loan in one of the 2 ways.Those 2 ways are illustrated below-

Installment Repayment

1.If a user applies for an installment repayment plan then the user will have to repay the loan amount with 6% interest within 6 months. Let's assume,the user has taken a loan of tk 10000.So he has to repay 10600 tk within 6 months. So he will pay 1765 tk in each month.

2.If he is unable to repay his installments in any month,a penalty of 1% of his total loan amount will be added as a fine.

Bullet Repayment

1.If the user applies for bullet repayment,he will have to pay the whole loan amount after 6 months along with 5% interest. Let's assume,the user has taken a loan of tk 10000.So he will have to repay 10500 tk after 6 months.

2.If he is unable to repay his bullet payment within due time, a penalty of 5% will be added as a fine every month.

Phase 6-Profit Distribution

In this phase we will discuss how the profit will be distributed among the investors at the year end.

When a borrower will repay his loan amount(including interest and fine),the amount will be stored in A2F's wallet.At the year end,all investors will get back their invested money along with some profits.The profit amount for each investor will be set by his investment percentage of A2F total investment. Let's assume,there are 4 investors who invested 10000,20000,30000,40000 tk respectively. At year end A2F collects 1.2 lakh as a total amount. Then the

investor will get 12000,24000,36000,48000 tk respectively. Those amounts will be transferred to their app wallet.

Additional Phases

In this phase we will look at all other features of A2F.This will include description of phases named-Fundraising,User Statistics,System Statistics and User Rating. Details of every phase is given below-

Fundraising

1.When a user needs funds for medical purposes he will apply for fundraising and upload necessary documents and will mention a date within which he needs the money.
2.Those documents will be verified by authority.
3.After successful verification,a fundraising post will be shown in the dashboard of all users.
4.User's will donate money in response to that fundraising post.Those funds will be stored in the app wallet.
5. 2 days before the mentioned date our agent will get a notification. Then he will perform a logical transaction and also transfer the physical money to the bKash of that user.

User Statistics

Users can see their transaction histories as table format and also as a histogram.A2F will predict the loan amount as well as loan taking time by analyzing the user's previous transactions.

System Statistics

All transactions via A2F wallet will be shown to agents as well committee members using histogram. A2F will predict a probable money acquisition time and needed amount by analyzing all past transactions of A2F.

User Rating

Each user will be rated automatically as per transaction. For investing , the user will be rated through 5 stars, and the borrower will also be rated 5 stars if he returns on time. If the borrower fails to return on time then there will be a penalty and rating will be decreased.

5. Scenario Based Modeling

This chapter describes the Scenario Based Model for the A2F.

5.1 Introduction

When developing software, user satisfaction is given the highest priority. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Thus, requirements being with

scenario generation in the form of use cases, activity diagrams and swim lane diagrams.

5.2 Definition of use case

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of
circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users.

The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

5.3 Use case diagrams

Use case diagrams give the non-technical view of the overall system.

Level-0

Primary Actors: User (Agent,Lender,Borrower,Admin)

Secondary Actors: bKash, Email, SMS

Figure-1:Level 0 of A2F

Level-1

Primary Actors: User(Admin,Borrower,Lender,Agent)

Secondary Actors: bKash,Email,SMS

Figure-2:Level 1 of A2F

Description of level 1 :

1. **Account Management :** Users will login or sign up to the system. A user needs to provide Name, Institutional Email/University Registration ID,Phone No, Password, NID Card (Photo) to sign up. Whereas a student

needs to provide his name, email, institution, batch, roll and phone number for signing up. Users can log in using their email and password. Their emails must be verified while signing up.If a user forgets his or her password he or she can recover the password using email verification.

2. **Money Acquisition:** Money Acquisition is collecting funds for giving loans to borrowers. Admin will give the post for money acquisition and lenders will respond and invest money for a specific period of time.

3. **Transaction Management :** Logical and physical cash flow will be controlled here. Personal and app wallet adjustment will be done here.

4. **Loan Management:** If any borrower wants to take a loan will go to the borrow interface and upload a loan request form. Then this form will be accepted or rejected based on money reservation of the app. After getting approval the agent will send money via bKash , at the same time logical transaction will be performed.

5. **Repayment Management:** A borrower will select the repay method (installment / bullet) and repay accordingly. If he fails to repay then the system will set a penalty on the basis of loan amount.

6. **Fundraising:** Any user may apply for fundraising through uploading a fundraising request form. Admin will verify the request. If the request is real , the admin will give approval and the system will send this fundraising notice to every users' dashboard.

7. **Statistics:** Users' transactions will be shown by generating a histogram in users' dashboard. Every user will be able to see personal statistics. At the same time the system will generate the overall transaction of a month. This system statistics will be accessible to admin only.

Level-1.1: Account Management

Primary Actors: User
Secondary Actors: Email

Figure-3:Level 1.1 of A2F

Action & Response
A1: User will provide credentials to sign up.
R1: A OTP will be sent for verification to email
A2: User will provide OTP
R2: On successful providing of OTP, User will get a user registered notification

A1: User will provide username and password
R1: On providing correct credentials, user will be able to log in and directed to dashboard

A1: When user forgot the password, he will provide new password
R1: A OTP verification will take place and when successful the new password will get stored

Level-1.2: Transaction Management
Primary Actors: User,Admin
Secondary Actors: bKash,Email,SMS

Figure-4:Level 1.2 of A2F

Action & Response
A1: User will provide amount for cash in
R1: A physical transaction will take place using bKash webhook

A2: After successful transaction,A logical transaction will happen where the money will be added to the wallet of user
R2: User will be able to view the increased amount

A1: User will provide amount for cash out
R1: A notification will be sent to agent's dashboard
A1: A physical transaction will take place by our agent
R2: After successful transaction,A logical transaction will happen where the money will be deducted to the wallet of user

Level-1.3: Money Acquisition
Primary Actors: Lender,Admin

Figure-5:Level 1.3 of A2F

Level-1.4: Loan Management
Primary Actors: Lender,Borrower,Agent
Secondary Actors: bKash

Figure-6:Level 1.4 of A2F

Level-1.5: Repayment Management
Primary Actors: Borrower,Agent,Lender
Secondary Actors: bKash,Email,SMS

Figure-7:Level 1.5 of A2F

Level-1.6: Fundraising

Primary Actors: User
Secondary Actors: Admin

Figure-8:Level 1.6 of A2F

Level-1.7: Statistics

Secondary Actors: User,Admin

Figure-9:Level 1.7 of A2F

5.4 Activity Diagram

Activity Diagram Notation and Symbol

Activity Diagram of Level-1

Figure-1:Activity Diagram of Level-1

Activity Diagram of Level-1.1(Account Management)

Figure-2:Activity Diagram of Level-1.1(Account Management)

Activity Diagram of Level-1.2(Transaction Management)

Figure-3:Activity Diagram of Level-1.2(Transaction Management)

Activity Diagram of Level-1.3(Money Acquisition)

Figure-4:Activity Diagram of Level-1.3(Money Acquisition)

Activity Diagram of Level-1.4(Loan Management)

Figure-5:Activity Diagram of Level-1.4(Loan Management)

Activity Diagram of Level-1.5(Repayment Management)

Figure-6:Activity Diagram of Level-1.5(Repayment Management)

Activity Diagram of Level-1.6(Fundraising)

Figure-7:Activity Diagram of Level-1.6(Fundraising)

Activity Diagram of Level-1.7(Statistics)

Figure-8:Activity Diagram of Level-1.7(Statistics)

5.5 Swimlane Diagram

Level 1

Level 1.2 (Transaction Management)

Level 1.3 (Money Acquisition)

Admin
Lender
Posting Fundraising Notice
Call For Money
Invest

Level 1.4 (Loan Management)

Level 1.5 (Repayment Management)

Level 1.6 (Fundraising)

Level 1.7 (Statistics)

6. Class Based Modeling

6.1 Class Responsibility Collaboration

Class-based modeling identifies classes, attributes, and relationships that the system
will use. It represents the object. The system manipulates the operations. The elements of the class-based model consist of classes and objects, attributes, operations, and class – responsibility - collaborator (CRC) models. Classes are determined using underlining each noun or noun clause and entering it into the simple table. Attributes are the set of data objects that are defining a complete class within the context of the problem. The operations define the behavior of an object.

List of all nouns in sorted order

General Classification

Candidate classes are categorized based on the seven general classification. The analysis classes manifest themselves in one of the following ways:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

A candidate class is selected for special classification if it fulfills three or more characteristics.

Serial NO	Noun	General Criteria
1	Months	2
2	Admin	4,5,7
3	Agent	4,5,7
4	Authentication	2,3,7
5	Amount	2
6	App transaction history	3
7	App wallet	2,5,7
8	Borrow interface	7
9	Borrower	4
10	Borrower Wallet	2
11	Bullet repayment	3
12	bKash account	2
13	bKash wallet	2
14	bKash	1,7
15	Call for Money	3,7
16	Cash Flow Protocols	2
17	Cashout	3
18	Cash In	3
19	Circular	2
20	Committee member	2

21	Due time	2
22	Dashboards	7
23	Donation	2
24	Ending date	2
25	Existing amount	2
26	Fine	2
27	Fundraising	3,5,7
28	Fundraising post	2,3
29	Histogram	2
30	Institutional Email/University/Registration ID	2
31	Interest	2
32	Invest button	7
33	Invested money	2
34	Investment	2,3,7
35	Investment dashboard	7
36	Investment percentage	2
37	Investor	4
38	Installments	2
39	Ledger	2
40	Loan	2,3,7
41	Loan application form	2
42	Loan amount	2
43	Loan post	2

44	Loan repayment	3
45	Loan request status	2
46	Loan taking time	2
47	Login	3
48	Logical cash flow	7
49	Logical transaction	3
50	Last date	2
51	Maintenance fee	2
52	Necessary documents	2
53	Name	2
54	Normal user	4,5,7
55	Notification	2
56	Needed amount	2
57	NID Card (Photo)	2
58	Password	2
59	Password Recovery	3
60	Phone No	2
61	Physical cash flow	7
62	Physical transaction	3
63	Principal money	2
64	Probable money acquisition time	2
65	Profit amount	2
66	Profit distribution	3

67	Reason of loan	2
68	Repayment plan	2
69	Starting date	2
70	Sign up	3
71	System Statistics	2
72	Total amount	2
73	Total Investment	2
74	Transaction histories	3
75	Transactions	3
76	Transaction Completed	2
77	Threshold	2
78	User Rating	3,4,7
79	User stars	2
80	User Type	2
81	User Statistics	2
82	User wallet	2,5,7
83	Wallet types	2
84	Year end	2

Potential To Be Classes

1. Admin
2. Agent
3. Normal user
4. Authentication
5. App wallet

6. User wallet
7. bKash
8. Fundraising
9. Investment
10. Loan

Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria:

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

A candidate class generally becomes a class when it fulfills around three Characteristics.

Serial NO.	Noun	Selection Criteria
1	Admin	1,2,3,4,5
2	Agent	1,2,3,4,5
3	Normal user	1,2,3,4,5
4	Authentication	1,2,3,4,5
5	App wallet	4
6	User wallet	2,3,4,5
7	bKash	6
8	Fundraising	1,2,3,4,5
9	Investment	1,2,3,4,5
10	Loan	1,2,3,4,5

Selected Classes

1. Normal user
2. Admin
3. Authentication
4. Agent
5. bKash
6. App wallet
7. User wallet
8. Investment
9. Loan
10. Fundraising

1.Normal User Class

Attribute	***Methods***
+Usertype +Name +Institutional Email +University Registration ID +Phone No +Password +NID Card (Photo) +bKash Number	+edit information() +view fundraising posts() +view loan posts() +view call for money circular() +get transactions() +see loan prediction()
Responsibility	***Collaborations***
• Editing user information • Viewing loan posts, fundraising posts and call for money circular in the dashboard • Viewing all transactions in graph • Getting the prediction of loan amount and time	Authentication User wallet App wallet Loan Investment Fundraising

2.Admin Class

Attribute	***Methods***
	+User Type Change() +Create circular() +see system statistics() +see probable money acquisition prediction()
Responsibility	***Collaborations***
• Changing user types • Creating circular for call for money • Viewing of system's transactions in graph • Getting the prediction of probable money acquisition amount and time	Normal user App wallet Fundraising

3.Authentication Class

Attribute	***Methods***
	+signup() +login() +pass recovery()
Responsibility	***Collaborations***
Creating a new user Logging in an existing user Recovering a user's password	

4.Agent Class

Attribute	***Methods***
	+get physical transaction notification() -complete transaction()
Responsibility	***Collaborations***
Getting the notification of doing physical transaction Send back the response when the physical transaction is completed	

5.bKash Class

Attribute	***Methods***
	+make transaction()
Responsibility	***Collaborations***
Performing the 'cash in' using the bKash web hook.	

6.App Wallet Class

Attribute	***Methods***
App amount Needed amount CFM Starting date CFM Ending date Circular	+get circular() +save transaction() +get transaction() +send profit()
Responsibility	***Collaborations***
Providing the circular for call for money Saving a transaction to the database Getting data of any transaction from the database Distributing the profit at year end	

7.User Wallet Class

Attribute	***Methods***
User amount Total investment Total loan Received Donation amount	+cashin() +cashout() -save transaction()
Responsibility	***Collaborations***
Performing the logical cash in. Performing the whole cash out process. Saving a user transaction at the database	Bkash App wallet Agent

8.Investment Class

Attribute	***Methods***
Investment ID Investment type Amount Profit amount Remaining amount Return date	+make investment in CFM() +lend a loan()
Responsibility	***Collaborations***
Investing for call for money Lending for a loan	

9.Loan Class

Attribute	***Methods***
+ +loan amount +Reason of loan +Repayment type +Repayment amount -Loan submission status -remaining activation days +collected amount -threshold -last date of repayment -fine	+apply() +accept() +loan post() +increase collected amount() +repay loan()
Responsibility	***Collaborations***
Application of a loan Providing a loan post Distributing loan post to the dashboard of all users Disbursement of loan Repayment of loan	User wallet Agent

10.Fundraising

Attribute	***Methods***
Documents Last date Verification status	+apply +verification response +verify +fundraising post +donate +send donation amount
Responsibility	***Collaborations***
Applying for a fundraising Verifying the document Sending the verification response to user Creating and distributing fundraising post in the dashboard of users Donating on a fundraising post Sending donation amount to the applicable user	User wallet App wallet Agent

CRC Diagram

Authentication
Investment
Fundraising
Loan
App wallet
Normal User
User wallet
bKash
Admin
Agent

7. Behavioral Modeling

The behavioral model indicates how the software will respond to external events or
stimuli. In the context of behavioral modeling, two different characterizations of states must be considered: (1) the state of each class as the system performs its function and (2) the state of the system as observed from the outside as the system
performs its function.

7.1 State Transition Diagram

One component of a behavioral model is a UML state diagram that represents active states for each class and the events (triggers) that cause changes between these active states.
State Transition Diagram represents active states for each class of events (triggers). For this we identified all the events, their initiators and collaborators. In the State
Transition Diagram the states are shown in boxed texts, and the transition is represented by arrows. It is also called State Chart or Graph. It is useful in identifying
valid transitions.

7.1.2 Identifying Events

Events	Initiator	Collaborator
Sign Up	Normal User	Authentication
LogIn	Normal User	Authentication
Provide Information	Normal User	
Password Recovery	Normal User	Authentication

Change User Type	Admin	Normal User
Cash In	Normal User	User Wallet, App Wallet, bKash
Cash Out	Normal User, Agent	User Wallet, App Wallet
Collecting Money	Admin	Normal User
Call for Money	Admin	Normal User
Lend to Normal User	Normal User	Normal User
Create Circular	Admin	Normal User
Adjust Investment	User Wallet, App Wallet	Investment
Adjust Loan Amount	User Wallet, App Wallet	Loan
Fill up Loan Application	Normal User	Loan
Define Repay Method	Normal User	Loan
Get Penalty	Normal User	Loan
Get Loan Submission Status	Normal User	Loan
Get Accepted Loan Request Notification	Agent, Normal User	Loan
Get Back Invested Money	Normal User	App Wallet
Get User Statistics	Normal User	
Get System Statistics	Admin	
Apply for Fundraising	Normal User	Fundraising

ID-1:

Name: General User

ID-2:

Name: Authentication

ID-3:

Name: Admin

ID-4:

Name: User Wallet , App Wallet

ID-5:

Name: bKash

ID-6:

Name: Fundraising

7.2 Sequence Diagram

The second type of behavioral representation, called a sequence diagram in UML,
represents how events cause flow from one object to another as a function of time. The sequence diagram is a shorthand version of the use case. It represents vital classes and the events that cause behavior to flow from class to class.

7.2.1 Purpose of Sequence Diagram

- High-level interaction between active objects in a system
- The interaction between object instances within a collaboration that realizes a
  use case
- The interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

In our A2F project, we have to show the interactions a user goes through while surfing the whole system from start to finish and organize them according to the segmentation of time. It can be useful to represent the details of UML use cases, to see how objects and components interact with each other to complete a process. Moreover, it will help us to plan and understand the detailed functionality of the existing scenario that we are exploring.

ID-1:

Nae: Account Management

ID-2:

Name: Transaction Management

ID-3:

Name: Money Acquisition

ID-4:

Name: Loan Management

ID-5:

Name: Fundraising

8. Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

DFD of Level-0

Figure-1: DFD of Level-0

DFD of Level-1

Figure-2: DFD of Level-1

DFD of Level-2

Figure-3: DFD of Level-2

9. Entity Relationship Diagram

Definition of ER Diagram An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

Data Modeling Concept:

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

Data Objects :

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

Data object identification:

- User
- Depositor
- Borrower
- CFM
- System Wallet
- Loan
- Distribution
- Loan Request
- Repayment

ER Diagram:

Definition of ER Diagram An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

Schema Diagram:

Data Object	Attributes	Type Size
User	-UID -University RegID -Usertype -Username -Password -Phone Number -bKash Number -wallet -email -current year	String String String String String String String Number String String
Depositor	-amount contributed	Number
Borrower	-Applied Loan -Accepted Loans	Loan Loan
CFM	-CFMID -CreatorID -Starting Date -Ending Date -Date Remaining -Requested Amount -Satisfied Amount	String String Date Date Number Number Number
System Wallet	-WalletID -amount	String Number
Loan	-LoanID -Loan Status -Loan EMI -Recipient -Satisfied Amount	String String Number String Number
Distribution	-UID -walletID -AgentID -Withdrawal amount -Repayment Received	String String String Number Number
Loan Request	-UID -Loan ID -Requested Amount -Requested Date	String String Number Date

	-Request Activation Days	Number
Repayment	-LoanID -WalletID -Paid Amount -Loan Repayment Days -Remaining Repayment days	String String Number Number Number