THE AMERICAN UNIVERSITY IN CAIRO
الجامعة الأمريكية بالقاهرة

Software Design Document (SDD)

For “***Gam3ya***” application

Mohamed Elatroush 900152131

Omar Radwan 900153452

Ahmed Ibrahim 900160265

Mohamed Selim 900153178

Omar Kamaly 900160151

<Software Company>

Date: (11/28/2019)

Table of Contents

1. Introduction

1.1 Purpose

This software design document documents the architecture and system design of our Gam3aya application. This document is important to all the developers who will be working on developing/implementing the system, as well as the project manager. This document includes the data design, architecture design, interface design, and procedural design.

1.2 Scope

The intent is to elevate the Gam3aya service to a worldwide application with outstanding global reach. Our main objective is to allow the users to interact with each other for the purpose of securing large sums of money every periodic duration of time to support their respective needs. Every month, all the users except for the receiving user will pay an amount of money in order to accumulate a large quantity of currency that will end up being provided to the receiving user. Gam3aya service aims to aid individuals suffering from low financial income to gain desperately-needed short-term additional money to boost their jobs in order for their long-term dream goals and projects to be fulfilled.

1.3 Overview

Provide an overview of this document and its organization.

Section number	Key	Description
2	**SYSTEM OVERVIEW**	A brief description of the whole system and how it works.
3	**SYSTEM ARCHITECTURE**	A detailed description of the components within our system, also it contains how these components interact with each other.
4	**DATA DESIGN**	Explains how the data will be stored inside the system, as well as the class diagram of the whole system
5	**COMPONENT DESIGN**	Shows the responsibility of each component in the system, and the relation between the layers and sublayers of the architecture design.
6	**HUMAN INTERFACE DESIGN**	Contains explanation about the user interface and what will happen if a user presses on a specific button, as well as UI screenshots and a brief explanation of each.

1.4 Definitions and Acronyms

SRS: Software Requirements Specification

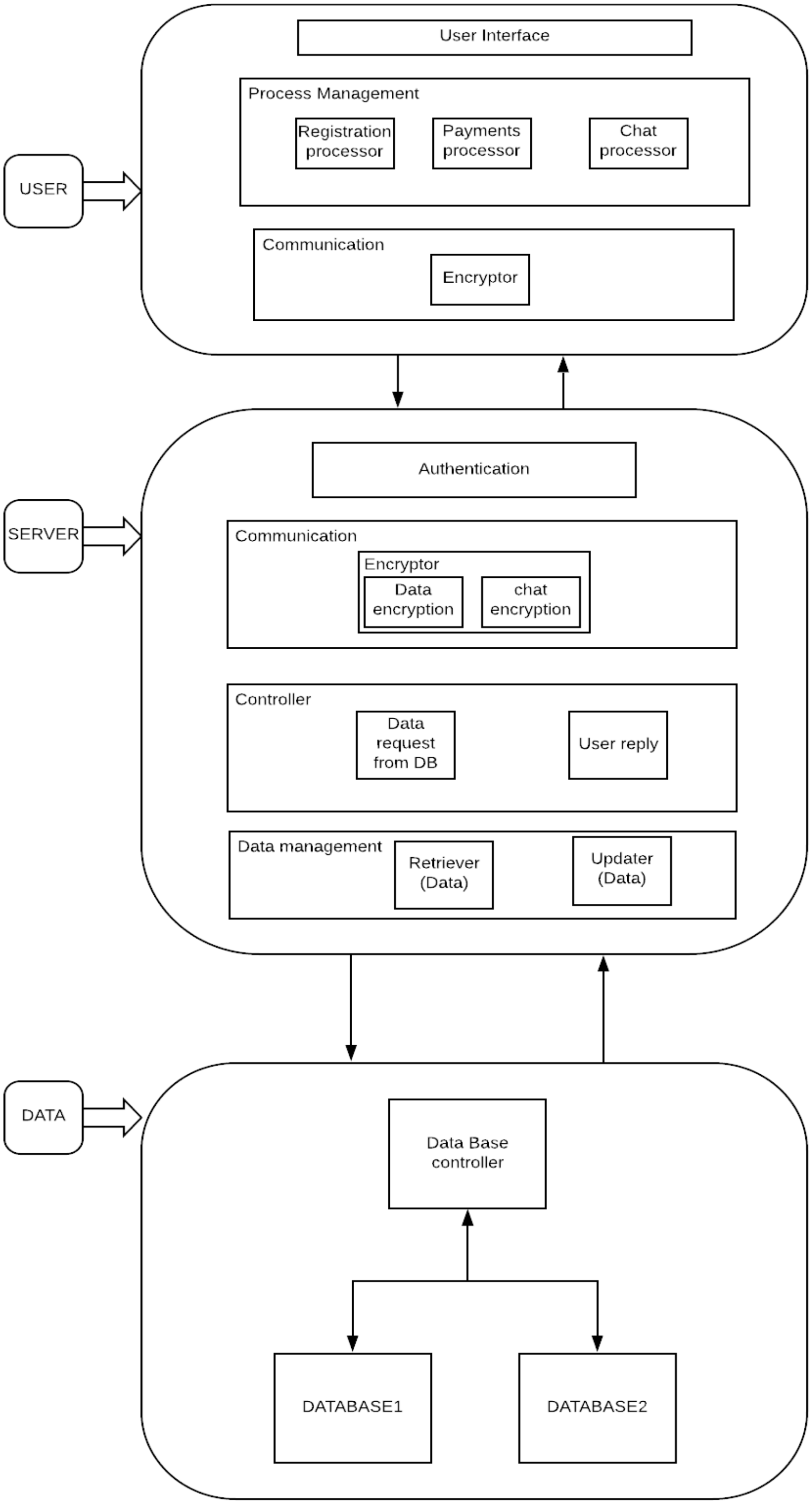
SDD: Software Development Document

2. System Overview

Firstly, when the user downloads the app from the respective app-store for the first time (Google Playstore for Android environments and Apple App Store for iOS environments), and decides to startup the app, he will be asked to either Log In or Register if the user already has an account. Once the user creates an account and is already logged in, the user is capable of either creating a brand-new Gam3aya or joining an already created one, also the user can utilise the search bar to search for other Gam3ayas and other users. When a user asks for permission to join a certain Gam3aya, a vote will occur in that Gam3aya to decide whether or not to accept/reject the new-coming user. This decision will most probably be decided by a handful of key characteristics attributed to the new-coming user such as age, background, profile and rating score. Gam3aya users indefinitely possess the ability to rate other members in a Gam3aya group, this is a fundamental right of every user and is viewed as a crucial procedure in order to determine who is worthy of taking advantage of the service's offerings. Users can increment the money on their virtual wallet for future payments via the list of payment methods supported. Any user will have the ability to delete their own account, as long as all the due payments are fully paid and he's not currently participating in an active Gam3aya. When a user first creates a new Gam3aya he is initially considered as the only admin of the group, at that instant they can modify the various settings of that Gam3aya (duration, limit..etc.). When the aforementioned user decides to invite others to join his created Gam3aya, the other members will have the exact permissions given to the first creator of the group (the starting user). This is crucial to prevent any user taking advantage of a "honeypot" gam3aya that attracts individuals for its high payout only to be shut down shortly after. Members cannot remove other members from the Gam3aya group by a simple click, a majority vote must first occur to enquire the whole group whether or not that specific member should be expelled. The user will not be removed if he has not paid yet, as the duration of Gam3aya is still active and it will simply be fraud at this point. If all members collectively wish to remove another member as soon as possible, because he/she doesn't pay on time and generally lacks cordiality despite maintaining a high score rating previously, members could report him to the allocated system administrators. Every member within a group preserves his right to construct a vote determining the set of criteria for a group, for example who will get paid first . This vote creation however is limited to one attempt per week. Inactive Gam3ayas could also be deleted by system admins. The application only takes a registration fee from the users as a method for profit.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design



3.2

Decomposition Description

User layer:

Layer	Component	Description
User Interface	User Interface	The space where the user and system interacts with each other. The use of the input device and the software.
Process Management	Registration processor	Takes the user registration credentials as an input and process this data and authenticates it (works with logging in or signing up).
	Payments processor	Takes the user payment information and responsible for transferring/receiving money from the user (To pay for other Gam3ayas or add money to the user's wallet).
	Chat processor	Responsible for the communication function that allows the user to chat with the group members/general chat.
Communication	Encryptor	Used for encrypting/decrypting important information to prevent any kind of harmful attacks on the system.

Server layer:

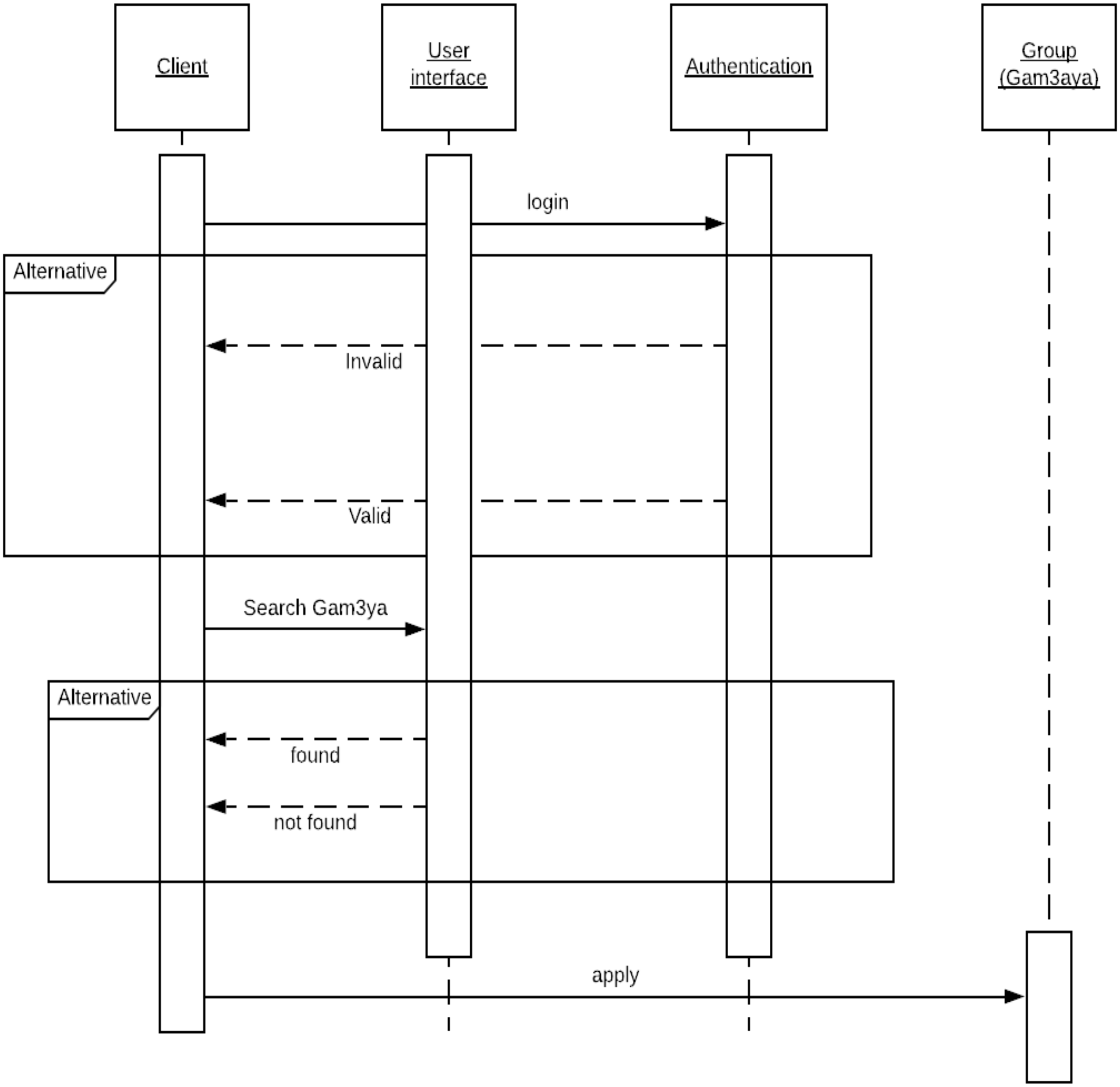
Layer	Component	Description
Authentication	Authentication	Used for authenticating and verifying users' credentials while logging in or signing up for the first time.
Communication	Data encryption	Used for encryption/decryption of data to protect users' information from external attacks.
	Chat encryption	An encrypting/decrypting module to protect the users' private, when they communicate in private chats (as they can exchange private information about their payments..etc.).
Data Management	Retriever (Data)	Concerned with communication with the database containing the data.
	Updater (Data)	Communicates with the database and responsible for updating it after the occurrence of a modification in the database.

Data layer:

Layer	Component	Description
Database controller	Database controller	It has the ability to select

		which database layer to use. It’s responsible for receiving/sending data to the server layer.
DB	Database 1	To avoid data inconsistency both databases(1 & 2) are the same to guarantee consistency.
	Database 2	To avoid data inconsistency both databases(2 & 1) are the same to guarantee consistency.

Payment sequence diagram:



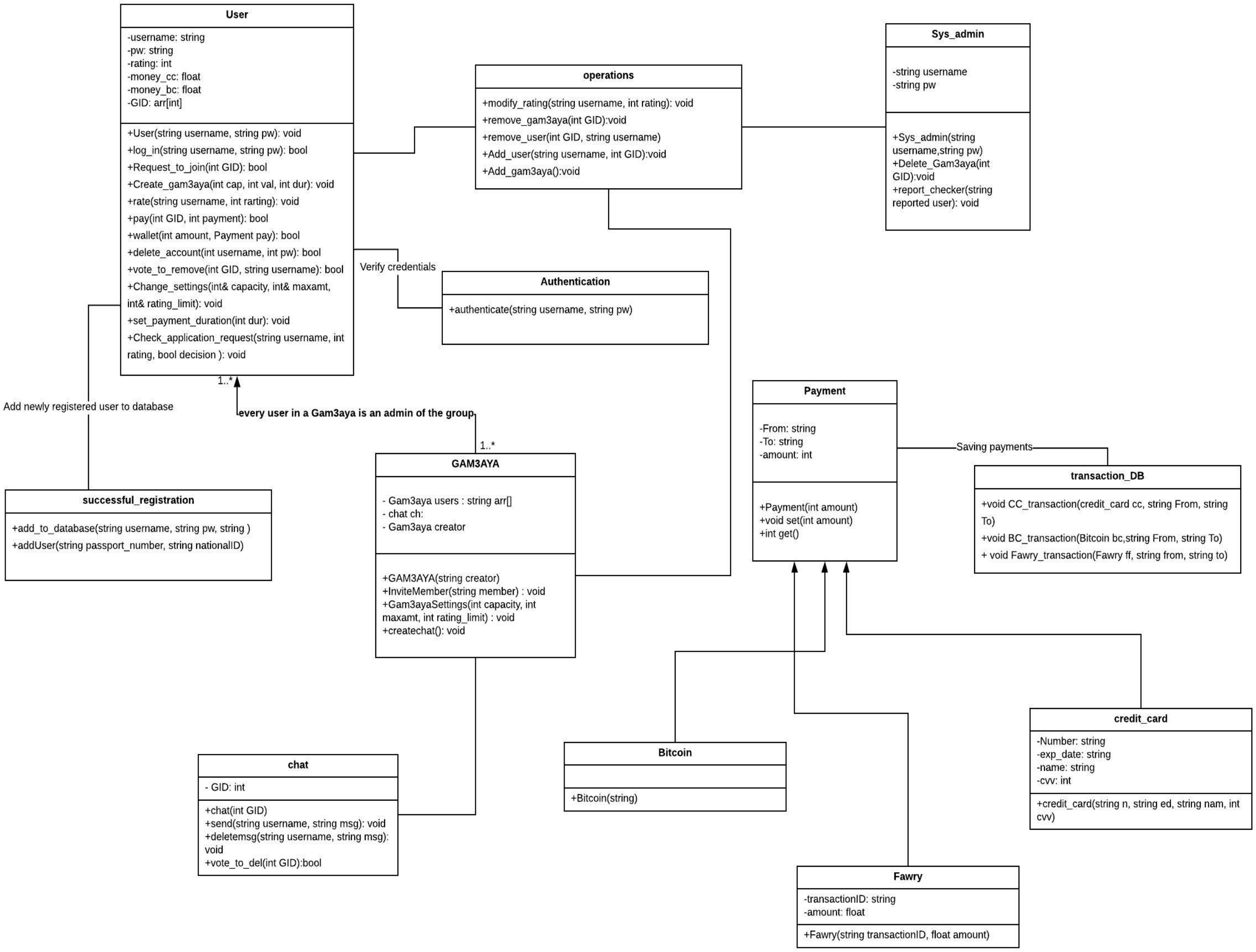
3.3 Design Rationale

The system architecture is a combination of both Layered and Client-server architectures. This choice shows how the system components interact with each other. Choosing different system architectures and combining them together extracts the advantage of each architecture and combine these advantages together. We’ve chosen 2 DBs, one is a copy of the other one to guarantee consistency, and also if an error occurs within a database we can switch to the second DB as it’s a copy of the first one, excluding the error that happened, to avoid system damage.

4. DATA DESIGN

4.1 Data Description

The data is stored in defined tables, these tables have the ability to be updated, insert items in the table, and delete data from the table when given the order to do so. The class diagram of our system is shown below, as it explains the classes that will be used within the system.


User
-username: string
-pw: string
-rating: int
-money_cc: float
-money_bc: float
-GID: arr[int]
+User(string username, string pw): void
+log_in(string username, string pw): bool
+Request_to_join(int GID): bool
+Create_gam3aya(int cap, int val, int dur): void
+rate(string username, int rarting): void
+pay(int GID, int payment): bool
+wallet(int amount, Payment pay): bool
+delete_account(int username, int pw): bool
+vote_to_remove(int GID, string username): bool
+Change_settings(int& capacity, int& maxamt, int& rating_limit): void
+set_payment_duration(int dur): void
+Check_application_request(string username, int rating, bool decision): void
operations
+modify_rating(string username, int rating): void
+remove_gam3aya(int GID):void
+remove_user(int GID, string username)
+Add_user(string username, int GID):void
+Add_gam3aya():void
Sys_admin
-string username
-string pw
+Sys_admin(string username,string pw)
+Delete_Gam3aya(int GID):void
+report_checker(string reported user): void
Verify credentials
Authentication
+authenticate(string username, string pw)
1..*
Add newly registered user to database
every user in a Gam3aya is an admin of the group
1..*
Payment
-From: string
-To: string
-amount: int
+Payment(int amount)
+void set(int amount)
+int get()
Saving payments
transaction_DB
+void CC_transaction(credit_card cc, string From, string To)
+void BC_transaction(Bitcoin bc,string From, string To)
+ void Fawry_transaction(Fawry ff, string from, string to)
GAM3AYA
- Gam3aya users : string arr[]
- chat ch:
- Gam3aya creator
+GAM3AYA(string creator)
+InviteMember(string member) : void
+Gam3ayaSettings(int capacity, int maxamt, int rating_limit) : void
+createchat(): void
successful_registration
+add_to_database(string username, string pw, string)
+addUser(string passport_number, string nationalID)
chat
- GID: int
+chat(int GID)
+send(string username, string msg): void
+deletemsg(string username, string msg): void
+vote_to_del(int GID):bool
Bitcoin
+Bitcoin(string)
credit_card
-Number: string
-exp_date: string
-name: string
-cvv: int
+credit_card(string n, string ed, string nam, int cvv)
Fawry
-transactionID: string
-amount: float
+Fawry(string transactionID, float amount)

4.2 Data Dictionary

Class Name	**User**	
Class ID	1	
Attributes	Username: string	Holds the user’s username
	Pw: string	Holds the user’s password
	Rating: int	Holds the user’s rating
	Money_cc: float	Holds the value of the user’s credit card
	Money_bc: float	Holds the value of the user’s bitcoin
	GID: arr[int]	It holds the ID of a Gam3aya, as each Gam3aya has a specific GID.
Methods	User(string username, string pw)	The constructor of the class User, it contains the username and the password that the user entered, and it verifies them.
	log_in(string username, string pw): bool	Responsible for authenticating the entered user credentials, the boolean return is true when he’s allowed to log in and false otherwise.
	Request_to_join(int GID): bool	When a user applies to join a specific Gam3aya this function is called, then a notification will be sent to all the users inside that Gam3aya, a voting will occur to accept/reject the applicant
	Create_gam3aya(int cap, int val, int dur): void	Create Gam3aya function is responsible for creating a new gam3aya group. It takes the inputs: cap(maximum allowed members of the group), val(money to be distributed at the end of each month on one member) and dur(duration of a particular gam3aya group)

	rate(string username, int rating): void	A function called whenever a user chooses to give a rating to another user. The other user’s username is passed as an argument to that function.
	pay(int GID, int payment): bool	Function called whenever a user attempts to pay. Gam3aya ID is passed as a parameter, and the amount to be paid.
	wallet(int amount, Payment pay): bool	A function called whenever the user tries to add funds to his virtual wallet. The amount to be paid is passed as a parameter)
	delete_account(int username, int pw): bool	The delete account method is called when the user decides to remove his account. The parameters passed are the username and password, then within the function several checks are done to whether return true or false.
	vote_to_remove(int GID, string username): bool	Any member can request a vote to all other Gam3aya members to remove another member.
	Change_settings(int& capacity, int& maxamt, int& rating_limit): void	Any user within a Gam3aya can use this function. The user can change the settings such as the capacity of a group, amount to be paid at the end of each month and a minimum rating limit.
	set_payment_duration(int dur): void	A user in a Gam3aya group can change the duration of payment. The function takes the original duration as an input and adds the new duration to it.
	Check_application_request(string username, int rating, bool decision): void	Any user can check a request from another user, and if the number of users within a group exceeds 50% the applicant will be accepted. It’s called when any member of the Gam3aya decides to view Gam3aya requests.

Class Name	**Authentication**	
Class ID	2	
Attributes	-	
Methods	authenticate(string username, string pw):bool	This function is responsible for verifying the credentials of the user who is trying to login with his registered username and password. It returns true if they were both entered correctly and false otherwise.

Class Name	**Gam3aya**	
Class ID	3	
Attributes	Gam3aya users : string arr[]	This array contains the usernames of every user within a Gam3aya.
	chat: ch	ch is of type chat. For communication between users.
	Creator: string	This contains
Methods	GAM3AYA(string creator)	This is the constructor of class Gam3aya, and will be called when a new Gam3aya group is created. It takes the creator of it as a parameter.
	InviteMember(string member) : void	This method will be called whenever any member of the group decides to add an external member to that group. A vote will occur to decide.
	Gam3ayaSettings(int capacity, int maxamt, int rating_limit) : void	Any member can use this method, as long as every other member agrees to that change.

	createchat(): void	When any member decides to create a chat, this function will be created to allow the communication between the users to happen.

Class Name	**chat**	
Class ID	4	
Attributes	GID: int	GID contains the ID of the Gam3aya that created a chatroom
Methods	chat(int GID)	When Gam3aya member decides to create a chat, this constructor will be called carrying the ID of that Gam3aya.
	send(string username, string msg): void	When a member decides to send a text, the send method will be called containing the username of the sender and a string carrying the sender's text.
	deletemsg(string username, string msg): void	This method will be called when any member can delete a message he sent earlier.
	vote_to_del(int GID):bool	To delete the whole chat, a vote will happen, it will ask the user if they agree to delete that chat, therefore a boolean is return.

Class Name	5	
Class ID	**Payment**	
Attributes	From: string	This string stores the sender's username.

	To: string	This string stores the receiver’s username
	amount: int	The amount integer holds the money that the sender wants to transfer to the receiver
Methods	Payment(int amount)	When the user decides to make a payment using his credit card or Bitcoin account this method will be called.
	Void set(int amount)	Method called when the user decides to set the amount to be paid. The amount variable is passed as an argument to that function.
	Int get()	Called when the database tries to save the amount to be paid by the sender

Class Name	6	
Class ID	**credit_card**	
Attributes	Number: string	This variable holds the credit card number as a string.
	exp_date: string	This variable holds the exp date of the card
	name: string	This variable holds the card holder’s name
	cvv: int	This variable hold the 3 integer security code
Methods	credit_card(string n, string ed, string nam, int cvv)	(This function inherits from Payment class). Function called whenever the user decides to make a payment using the credit card

Class Name	7

Class ID	**Bitcoin**	
Attributes	No attributes	
Methods	Bitcoin(string)	Function that is called whenever the user chooses paying with Bitcoin method

Class Name	8	
Class ID	**Fawry**	
Attributes	transactionID: string	Variable that holds the transaction number
	amount: float	Variable that holds the amount to be transferred
Methods	Fawry(string transactionID, float amount)	This method is called whenever the user decides to pay using his Fawry account

Class Name	9	
Class ID	**transaction_DB**	
Attributes		
Methods	void CC_transaction(credit_card cc, string From, string To)	When the payment is performed successfully, this method will be called to insert the credit card payment in the transaction database. The details of the sender's and receiver's credit will be updated in the database.
	void BC_transaction(Bitcoin bc,string From, string To)	When the payment is performed successfully using the bitcoin transaction. Both senders and receivers credits will be updated in the database
	void Fawry_transaction(Fawry ff, string from, string to)	When the payment is performed successfully using Fawry

		transaction. Both senders and receivers will be updated in the database.

Class Name	10	
Class ID	**successful_registration**	
Attributes		
Methods	add_to_database(string username, string pw, string)	When the registration is successful, the username, pw and name will be stored in the database.
	addUser(string passport_number, string nationalID)	After adding to the database. User's passport number and national ID will be attached under his username.

Class Name	11	
Class ID	**operations**	
Attributes		
Methods	modify_rating(string username, int rating): void	When the user is rated by another user, this function will be called to update rated user's rating in the database
	remove_gam3aya(int GID):void	This method is called when a Gam3aya group is deleted. It removes the Gam3aya ID completely from the database
	remove_user(int GID, string username)	When a user is removed from a Gam3aya, this method is called. It removes that user from the Gam3aya DB using GID number.
	Add_user(string username, int GID):void	When a user joins a Gam3aya this method is called. This function inserts the user's details

		into the database
	Add_gam3aya():void	When a Gam3aya is created this method is called. It assigns a unique ID to the newly created Gam3aya and saves it in the database.

5. Component Design

This section is an extension to section 3, as it goes over the architectural design of the Gam3aya system. The three main layers in our architectural design are the client, server and data layers.

Client layer:

UI sublayer:

The client layer contains sublayers, one of those sublayers is the UI sublayer. This sublayer is concerned with offering the ability of interaction between the user and the application. The UI sublayer is concerned with displaying all the features of the Gam3aya system.

Process management sub-layer:

The next sublayer of the client layer is the process management sub-layer. This sub-layer is concerned with handling different features offered to the user such as: registration processor, payments processor and chat processor. Registration processor handles the registration process of the new users, as well as verifying the credentials of an already registered user whenever he/she tries to log in. The second feature handler is the payments processor, it's concerned with handling all payments transfers from the sender to the receiver and verifying that the correct amount was transferred successfully to the receiver. Finally, the chat processor feature is concerned with sending and receiving messages features, it contains filtering function to prevent bad language, as well as auto reporting the user, which might affect the user's rating without being reported by another human.

Communication sub-layer:

The final sublayer is the communication Layer, it contains the encryption feature, this feature is concerned with encrypting/decrypting the data being transferred from the client layer to the server layer. To avoid any kind of attacks on the system and the privacy of the users, this is when the encrypting module is introduced, to ensure the protection of the users of the system. The encrypted message can only be decrypted by the receiving end.

Server Layer:

Authentication:

Authentication sub-layer is concerned with handling user registration and log in, if a user registers for the first time it verifies every detail he entered while registering to the system such as the passport number and national, as well as verifying the user’s credentials when trying to log in into his/her account. For example, if a user tries to log in with his username and password, the system will automatically go and check the user’s username if it exists in the system’s database or not, if it does then the user could proceed and access his account, else an error message should pop asking the user to log in again.

Communication:

The communication sub-layer contains the encryption module which is responsible for encrypting/decrypting data to avoid any external attacks on the system, as our system is trying to guarantee full safety and protection to all the registered users. Information such as credit card numbers are very sensitive, therefore encrypting such information would make it impossible for the attackers to understand what is written when they hack the system. Finally the chat encryption feature is also used, because if two users are exchanging sensitive/private information, encryption would protect these private information from any external attackers on the system.

Controller sub-layer:

Controller sub-layer contains Data request from DB feature. This feature handles user action on Gam3aya’s application. The application will obtain the needed data from the database and apply the changes made by the user, then the modified data will be sent back to the database.

Secondly, the user-reply feature is concerned with ordering the data and generating these replies automatically to the user’s when they’re are done from a specific task.

Data Management sub-layer:

Retriever feature is concerned with the communication between the server layer and the data layer. The data receiver feature receives the needed information from the database and has the ability to modify it, then return it back to the database table.

Updater feature is used whenever the user issues a request changes to the database will occur therefore, the updater feature modifies this data and sends it back to the database after being modified.

Data layer:

______Database controller handles sending needed data from the database (data layer) to the server layer, possessing the ability to differentiate between the two databases that the system contains depending on the requested data.

DATABASE1 & DATABASE 2 sub-layers:

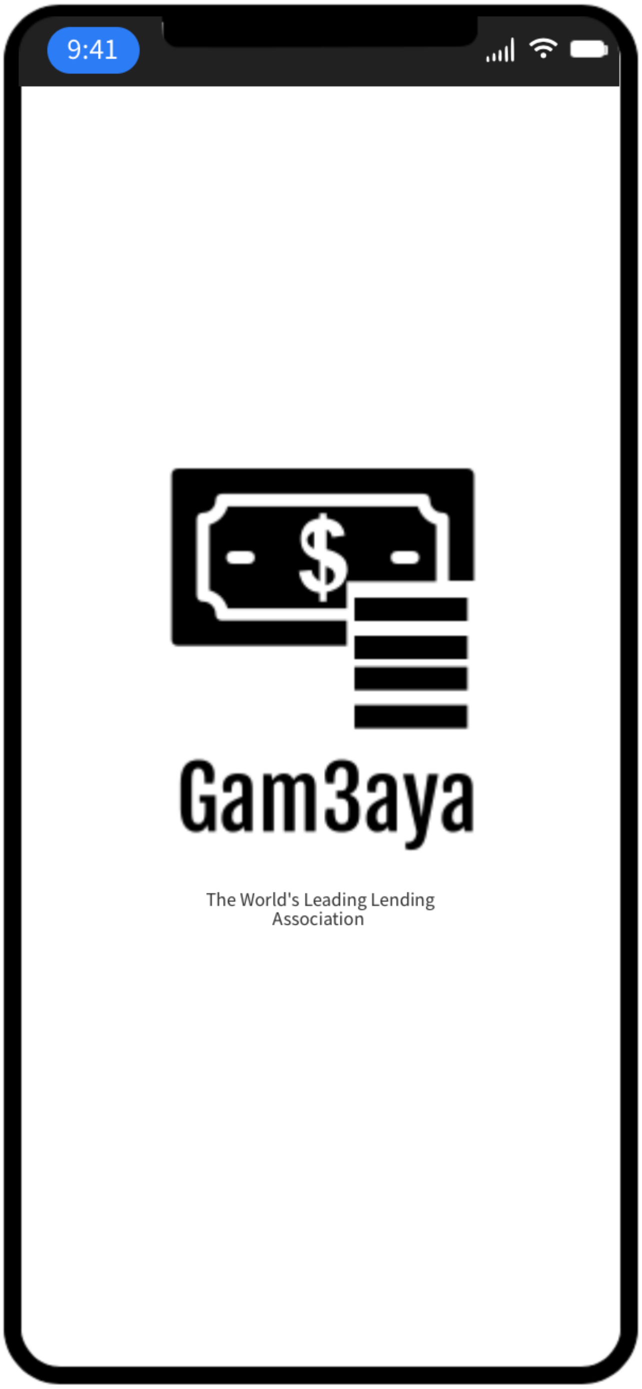
Both databases are the same to avoid failure of the system, since an error might occur in one of the two DBs, therefore as a safety measure the second DB would be useful to the system. The two DBs are always in sync, so if a modification or change occurs in one of the two databases, the second one will automatically copy the modified data, ensuring no discrepancies in data occur between both databases.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

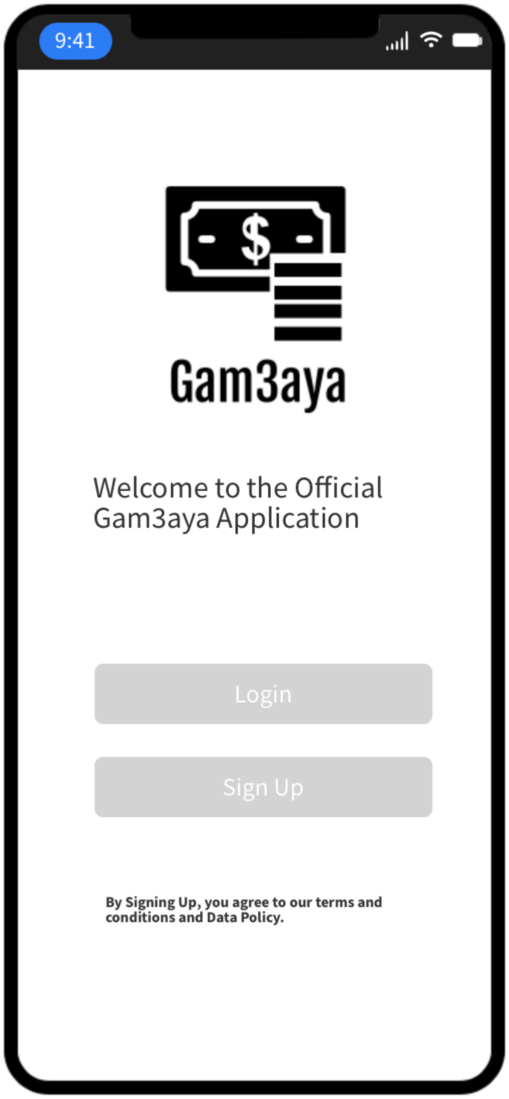
So this system provides the user with a friendly environment on a mobile application and web application. As soon as the user starts to access our website, he would have 2 options either sign up, that's the case if it's the first time for the user to open our website and wants to join Gam3aya, on the other hand ,if the user has already reached our website before and already joined us he will easily login. However, if he forgot his password, he will be able to change his password using his mobile phone or email address that were used in the sign up process. In the homepage of the application, the user will find all the necessary information; the gam3aya he's currently in, the payment dates , the number of points in order to track his points. The user can request to join a gam3aya or can create gam3aya with specific requirements or search for one as well. In addition to that, the user can change his payment method at anytime. We also have a contact page, where the user can find most of the questions answered in the FAQ questions or in our database, however, if not found he will have the option to contact us by email.

6.2 Screen Images



Gam3aya

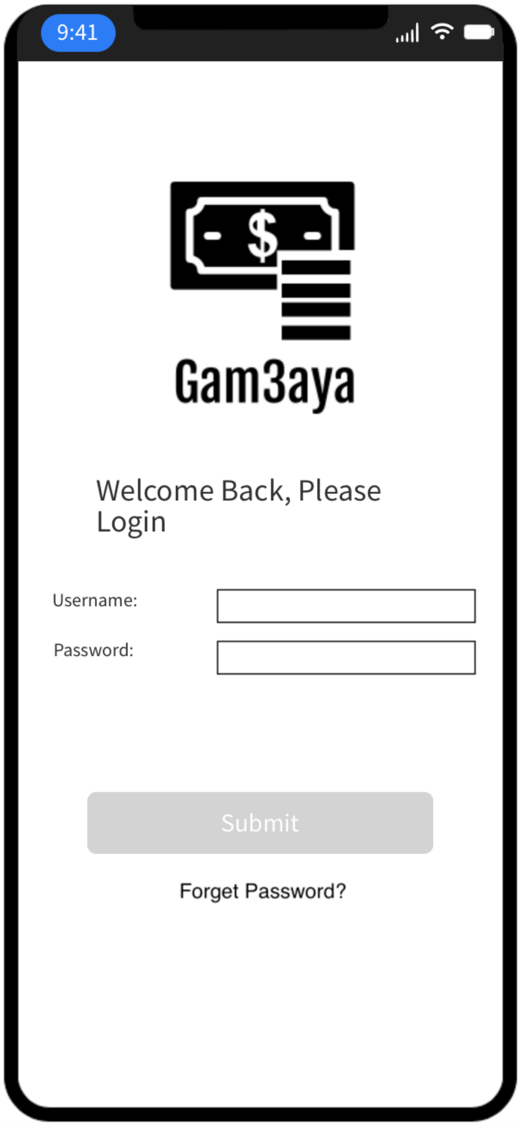
The World's Leading Lending Association



Welcome to the Official Gam3aya Application

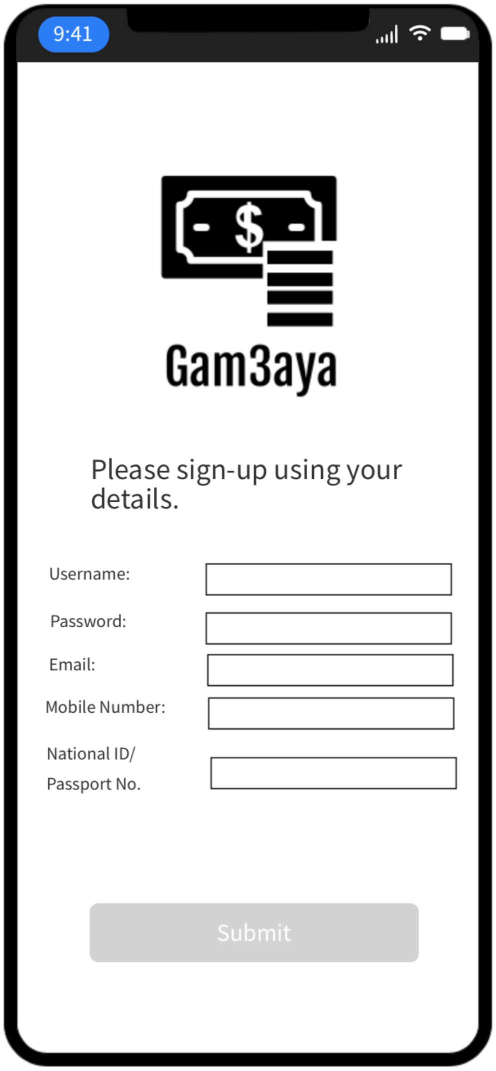
By Signing Up, you agree to our terms and conditions and Data Policy.

Element	**Action**
Login Button	Takes him to login page
Sign Up Button	Takes him to Sign Up page



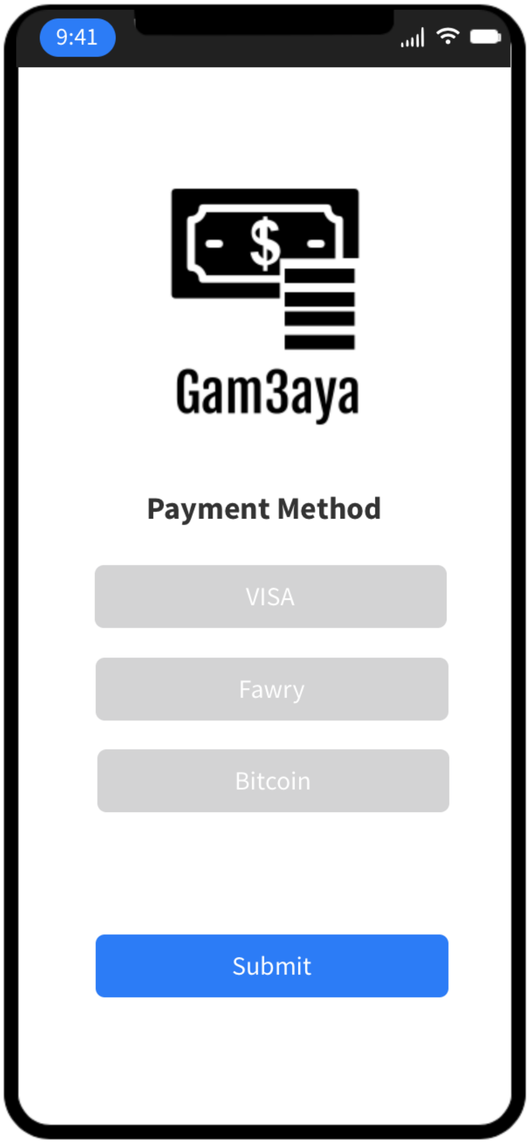
Welcome Back, Please Login

Element	**Action**
Username	Enter username
Password	Enter password
Submit Button	This button starts the login process
Forget Password	Button to help the user change his password

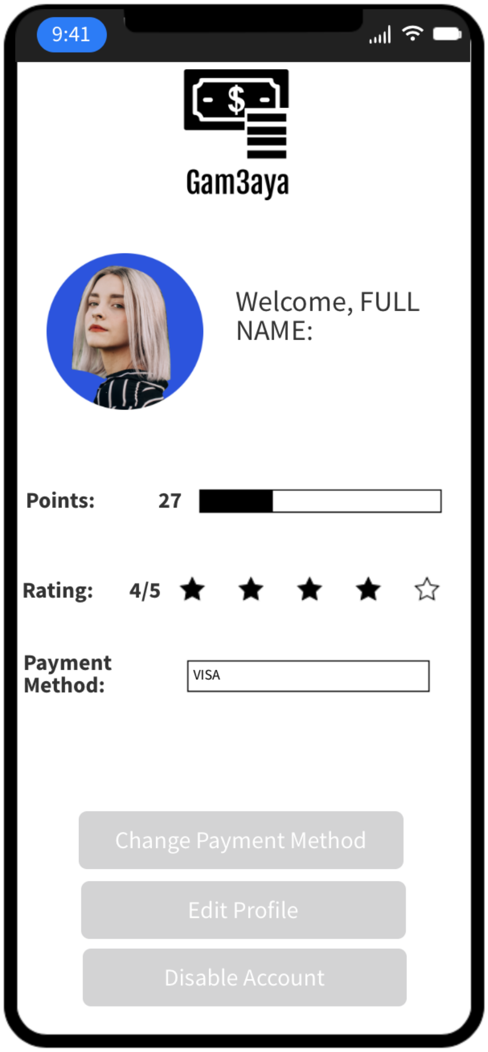


Please sign-up using your details.

Element	**Action**
Username	Enter username
Password	Enter password
Submit Button	This button starts the login process
Email	Enter email for this account
Mobile number	Enter mobile number for the account
National ID/ Passport	Enter id or passport number



Element	**Action**
VISA	Button to enter visa details
Fawry	To select farwy as payment method.
Bitcoin	To select bitcoin as payment method.
Submit	To finish payment method section



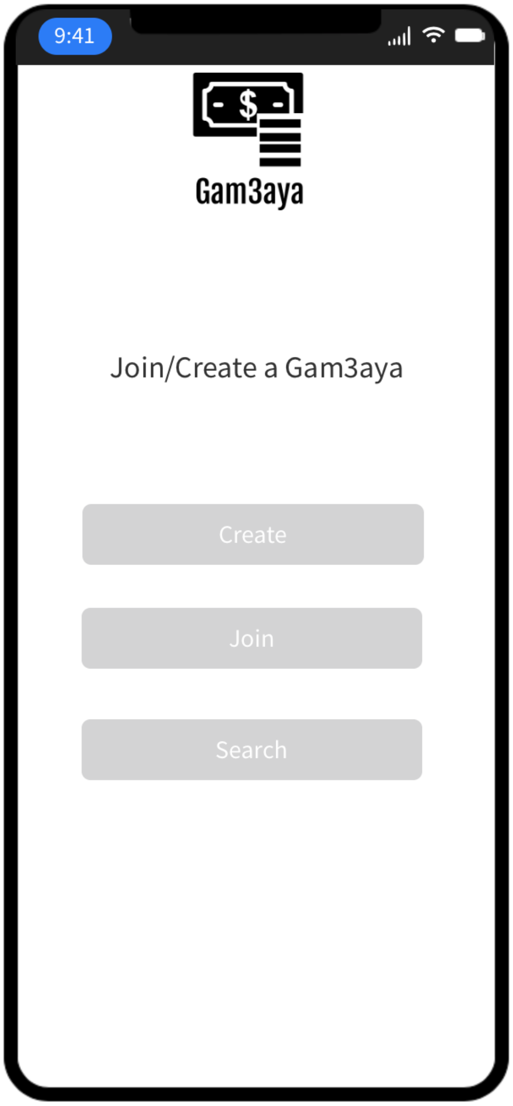
Welcome, FULL NAME:

Points: 27

Rating: 4/5 ★ ★ ★ ★ ☆

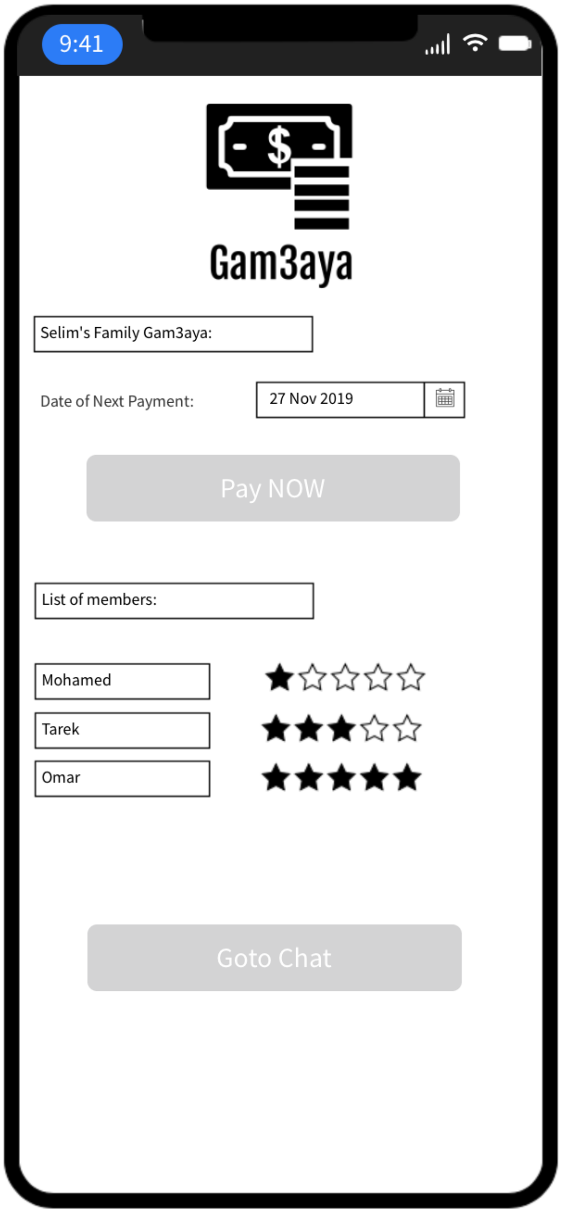
Payment Method: VISA

Element	Action
Points	View points of the user
Rating	Show the rating of the user
Payment method selected	Shows the current payment method
Change payment method	Button to change payment method
Edit profile	To edit something on the profile
Disable account	To delete the account



Join/Create a Gam3aya

Element	**Action**
Create	Create a gam3aya
Join	Join a gam3aya
Search	Search for a gam3aya



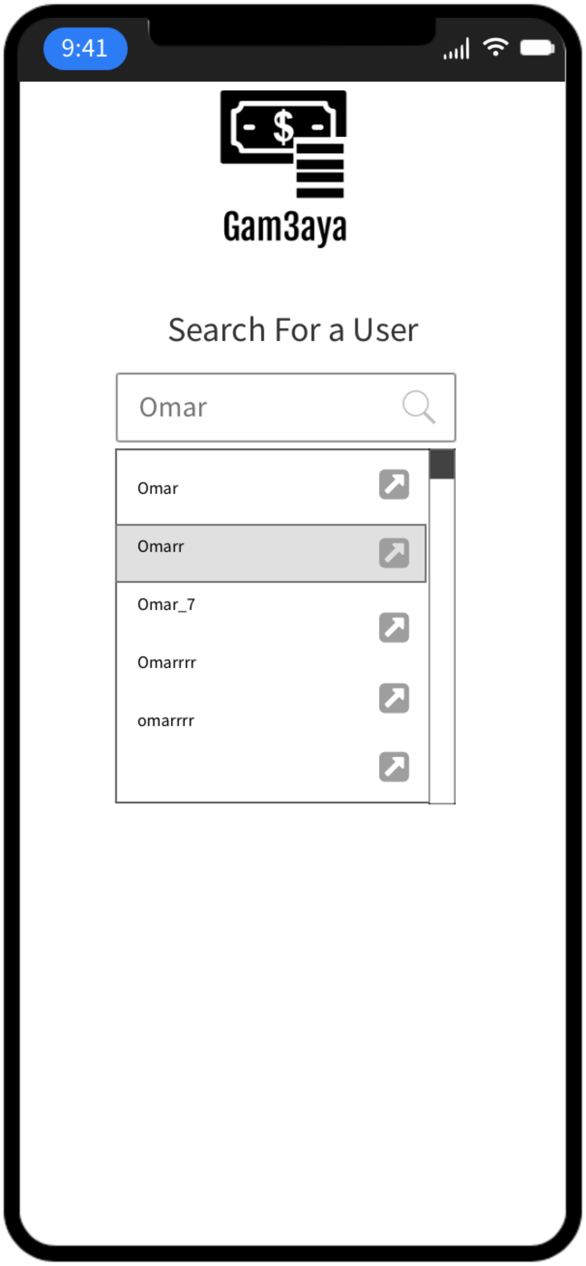
Selim's Family Gam3aya:

Date of Next Payment: 27 Nov 2019

List of members:

Mohamed	★☆☆☆☆
Tarek	★★★☆☆
Omar	★★★★★

Element	Action
Date of next payment	Shows user when is the next payment
Pay button	User can pay using this button
List of members	Shows the list of members and their ratings
Goto chat	Redirects the user to the chat room



Search For a User

Omar

Omar

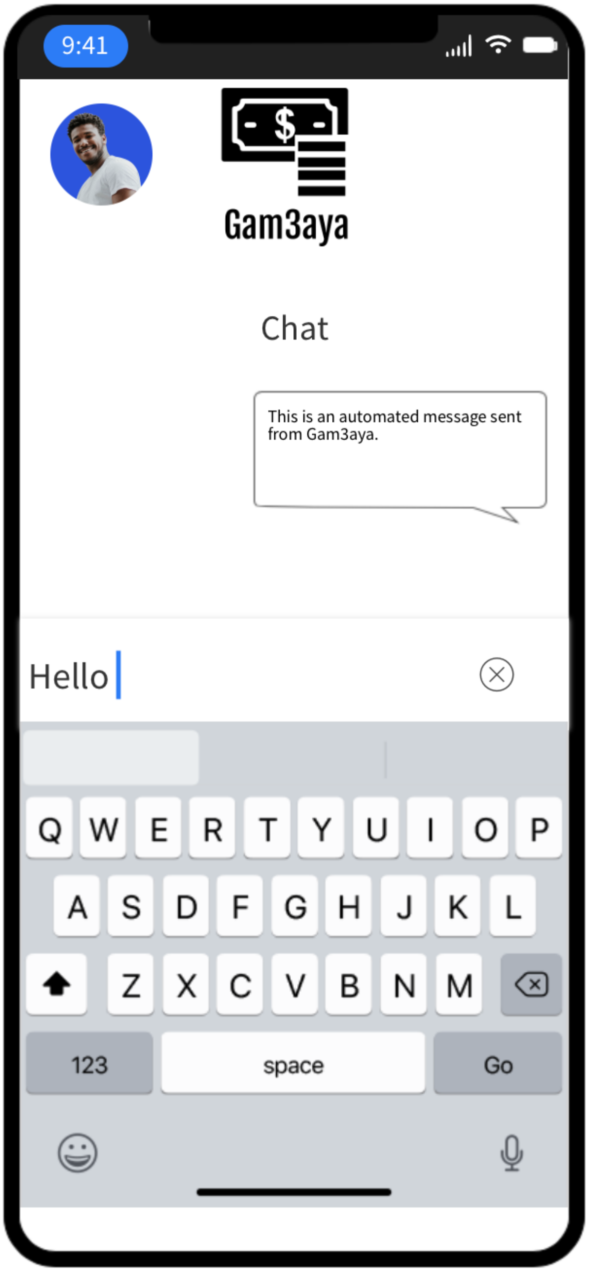
Omarr

Omar_7

Omarrrr

omarrrr

Element	Action
Search bar	Searching for other user with recommendations.



This is an automated message sent from Gam3aya.

Hello

Element	**Action**
Chat	Shows previous chat
Text	Type and send chat in this chat room.