

Table of Contents

1. Introduction	5
1.1. Purpose	5
1.2. Document Conventions	5
1.3. Intended Audience and Reading Suggestions	5
1.4. Product Scope	5
1.5. References	6
2. Overall Description	6
2.1. Product Perspective	6
2.2. Product Functions	6
2.3. User Classes and Characteristics	7
2.3.1. Gam3ya Participants	7
2.3.2. Financial agent	7
2.3.3. Customer service agent	7
2.3.4. System admin	8
2.4. Operating Environment	8
2.5. Design and Implementation Constraints	8
2.6. User Documentation	8
2.7. Assumptions and Dependencies	8
2.7.1. Assumptions:	8
2.7.2. Dependencies	9
3. External Interface Requirements	10
3.1. User Interfaces	10
3.1.1 Login/Sign up Interface	10
3.1.2 Find a Gam3ya Interface	11
3.1.3 Create A Gam3ya Interface	12
3.1.4 Gam3ya Details Interface	13
3.1.5 Forums Interface	14
3.1.6 Chat Interface	15
3.1.7 Help Interface	16
3.2. Hardware Interfaces	17
3.3. Software Interfaces	17
3.4. Communications Interfaces	17
4. System Features	18
4.1. Create an Account in Gam3ya	18

4.2. Check National ID	19
4.3. Verify Age Constraint	20
4.4. Login to Gam3ya	21
4.5. Use the tutorials	22
4.6. Request support	23
4.7. Search for a Gam3ya	24
4.8. Display Available Gam3yas	25
4.9. Join a Gam3ya	26
4.10. Create a Gam3ya	28
4.11. Check Reputation Points	29
4.12. Give Permission To Create a Gam3ya	30
4.13. Create a Gam3ya Forum	31
4.14. Create a Private Forum	32
4.15. Join a Private Forum	33
4.16. Join a Gam3ya Forum	34
4.17. Get Members' Approval	35
4.18. Complete a Gam3ya	36
4.19. Increase Reputation Points	36
4.20. Reduce Reputation Points	37
4.21. Send Message	38
4.22. ReceiveMessage	39
4.23. Pay the monthly amount	40
4.24. Receive payment	41
4.25. Manage the funds	43
5. Other Nonfunctional Requirements	44
5.1. Performance Requirements	44
5.2. Safety Requirements	44
5.3. Security Requirements	44
5.4. Software Quality Attributes	44
5.5. Business Rules	45
6. Other Requirements	46
Appendix A: Glossary	47
Appendix B: The proposed solution to the client's concerns	48
Appendix C: The client's interview questions	49

Appendix D: The use Case Diagrams	51
Create an Account	51
Payment subsystem	51
Chatting system	52
Create/Join a Gam3ya	52
The full Use Case Diagram	53
Appendix E: The Activity Diagrams	54
Create an account	54
Create a Gam3ya	55
Request Help	55
Join a Gam3ya	56

Revision History

Name	Date	Reason For Changes	Version
Gam3ya SRS	30/10/2019	Initial Draft	1.0
Gam3ya SRS	5/11/2019	Completed SRS Document	1.1

1. Introduction

1.1. Purpose

The purpose of this product is to create an online peer-to-peer lending software that implements the concept of Gam3ya securely. It also allows a degree of social interactions, including text messages, video and voice calls between users of the system. This document specifies the software requirements for the initial (first) version of the product as well as the proposed interfaces and their initial designs.

1.2. Document Conventions

- We use bold formatting for headings
 - Font size = 18 → heading 1: used for the headings of the 5 main pillars of the SRS document (Introduction, Overall Description, External Interface Requirements, System Features and Other Non-Functional Requirements)
 - Font size = 14 → heading 2: used for the headings of each subcomponent within each pillar in the SRS document
 - Font size = 12 → heading 3 used for further formatting of smaller sections within subcomponents
- We use a bolded asterisk followed by bolded sentence to represent a note to be considered
- Normal text has a font size of 11

1.3. Intended Audience and Reading Suggestions

This document will be submitted to various stakeholders that could be categorized into two groups of people. The first group of stakeholders contains the management; this includes Gam3ya owners, managers as well as project managers. This group would be concerned mainly with sections 1,2,4 and 5 in this document. They are required to verify product's purpose as well as its functional and non-functional requirements. The second group of stakeholders is mainly related to the technical side of the product; it includes the developers and testers and they would focus on sections 2, 3, 4 and 5. They are required to design the system in such a way that it meets the constraints, assumptions and dependencies described in section 2, the functional requirements described in section 4 and the non-functional requirements described in section 5 using the proposed software, hardware and communication interfaces explained in section 3.

1.4. Product Scope

The vision of this product is to allow peer-to-peer lending securely in a way that is available to a large amount of users. The main objective is to allow small fundings for individuals who don't have the ability to obtain a credit card and are of some financial need. Additionally, to help replace the traditional Gam3ya's with a cashless and more convenient and accessible way. Finally, though the product is virtual and solely accessible on the internet, it would still be able to allow social interactions between individuals participating in Gam3ya's in an incorporated online forum-like platform that allows messaging as well as voice and video calls.

1.5. References

Fawry's API documentation: <https://www.atfawry.com/developer-guide/>

Amazon Aurora documentation: https://docs.aws.amazon.com/rds/?id=docs_gateway

2. Overall Description

2.1. Product Perspective

The idea of the product originates from the traditional “gam3ya” that people in the Egyptian society partake in. Participants of a traditional “gam3ya” pay a certain amount of money periodically, collect the money and give it to one of the participants. There are no software systems that regulate the process. One of the participants is usually the “leader” of the gam3ya who collects and delivers money to other participants. However, The “gam3ya” product is a software that manages many gam3yas fairly and regulates the process of paying the participants according to their convenience. The product relies on external entities, mainly local payment service providers, to make sure that all the transactions are as secure as possible and to be easier to use. However, there are other mobile applications that lend money such as Earnin, DailyPay, and PayActiv. Despite the fact that the concept of a software lending money to the users is not brand-new, these applications are connected to the bank accounts of the user and money is taken from the user’s paycheck. Our product does not connect to a bank account of the user. The user is probably not expected to have a bank account (target audience). Therefore, the “gam3ya” product is software that replaces these types of application. However, the processes of payment and collecting money is distinct from all other similar applications.

2.2. Product Functions

The major functions Gam3ya will provide are the following:

- 1) *Register*: user can create an account
- 2) *Age validation*: check national ID to make sure that user is above 18
- 3) *Login*: prompt user to enter username and password to access their accounts
- 4) *Search for Gam3ya*: Allow high reputation users to search for their preferred gam3yas
- 5) *Create Gam3ya*: user can send a request to admin to create a gam3ya
- 6) *Join Gam3ya*: user can send a request to the gam3ya members to join the gam3ya
- 7) *Get Approval*: User receives an acceptance message to the gam3ya
- 8) *Send Message*: allow user to send messages to other participants
- 9) *Receive Message*: user can receive messages from other participants
- 10) *Pay Monthly Payment*: a user pays their monthly payment through one of the provided payment services

- 11) *Receive Payment*: user receives the amount of money they expect to receive
- 12) *Check Reputation points*: check user's reputation points before proceeding with any transactions
- 13) *Increase Reputation Points*: after finishing a Gam3ya, the user's reputation points increase
- 14) *Reduce Reputation Points*: whenever a user does not pay on time, their reputation points are reduced
- 15) *Use Help*: prompt user to request help from a customer service agent regarding the software usage.

2.3. User Classes and Characteristics

In Gam3ya there will be 4 different types of users:

2.3.1. Gam3ya Participants

This is the target end-user of Gam3ya and they can either join an existing Gam3ya or create a Gam3ya and wait for others to join. These will have the capability to enter a chat forum, send messages, view a Gam3ya, request to join a Gam3ya and create a Gam3ya. The system would categorize end-users in 2 groups

1. low-reputation user: a low reputation user is a user who is new to our system (and thus, we cannot trust them yet) or a user that doesn't deliver their payments on time. Low reputation users would only be allowed to join low-reputation Gam3ya's. Low reputations Gam3ya's allow a limited payment and only allow distribution of payments starting from the 7th month in order to reduce the amount of money lost in case someone refuses to pay after getting paid.
2. high-reputation user: a user that has participated in Gam3ya's in the system before and the log of their payments shows commitment and that due payments are paid on time. A high reputation user would be allowed to join high reputation Gam3ya's; these Gam3ya's follow the traditional Gam3ya format where payments collected from all users are distributed to a user starting from the 1st month.

***Extensive description of low-reputation and high-reputation users and Gam3ya's can be found in Appendix B**

2.3.2. Financial agent

A financial agent would be granted access to another level of information that Gam3ya participants do not have. In our system, they would be able to check users' payments as well as the system's bank account to verify whether those payments were delivered by the external service providers or not.

2.3.3. Customer service agent

This sort of users would be able to check users' logs and payments in order to answer any inquiries they might have; such inquiries could include questions about their reputation points and why it has decreased/increased.

2.3.4. System admin

Generally, the process of approval of a user's request to join or create a Gam3ya would be automated; however, for critical scenarios, an admin user will have to approve or reject the request manually; a critical scenario could include a user's request to create a Gam3ya with an extraordinarily large monthly payment that, according to their payment log, they might not be able to afford.

2.4. Operating Environment

The software can be used by anyone who has a smartphone. The software could be installed on iOS or Android. The software also will be implemented in the form of a website as well. The software depends on other software components such as systems of the local payment service providers (fawry, Vodafone cash, banking service, etc...) and cryptocurrency operators, such as Coinbase.

2.5. Design and Implementation Constraints

- Cryptocurrencies are not legalized in Egypt so we need official documents to allow the product to support trading in cryptocurrencies
- As a private organization, we will not be able to collect copies of participant's national IDs. Consequently, the validation process will be based on the ID number they input in the form which can be falsified.
- We will be relying on external payment service providers, such as Fawry and Vodafone Cash and therefore, some users might lack the knowledge of how to deal with such operators.

2.6. User Documentation

- Terms and conditions: users should agree on the terms of the system right after creating an account
- How to use Gam3eya: users who just registered will have a simple demo to show them the basic functionalities of the system
- Help Icon: the users can access "help" which basically has the user manual.

2.7. Assumptions and Dependencies

2.7.1. Assumptions:

- we will use Fawry's online API (we do not need any hardware interfaces with them)
- The "gam3ya" will have an account on each local payment service providers platform so that participants will transform to these accounts according to which system are they using

- The product will need “amazon aurora”: it is an online service provided by amazon that has database, cloud,...etc.
- Any participant of the gam3ya pays late, they will be paid late as well.
- The software will not display any advertisements for any other entities
- The identities of the users are not shown to each other. The user will enter a username that will be known with.

2.7.2.Dependencies

- The system will be entirely dependent on external payment providers and therefore, whenever their services are not available, our system will not be fully functional.
- Similar to the payment providers, our application and databases will be hosted on Amazon Aurora. Hence, any service shortage from Amazon would deem our system unusable during that time.

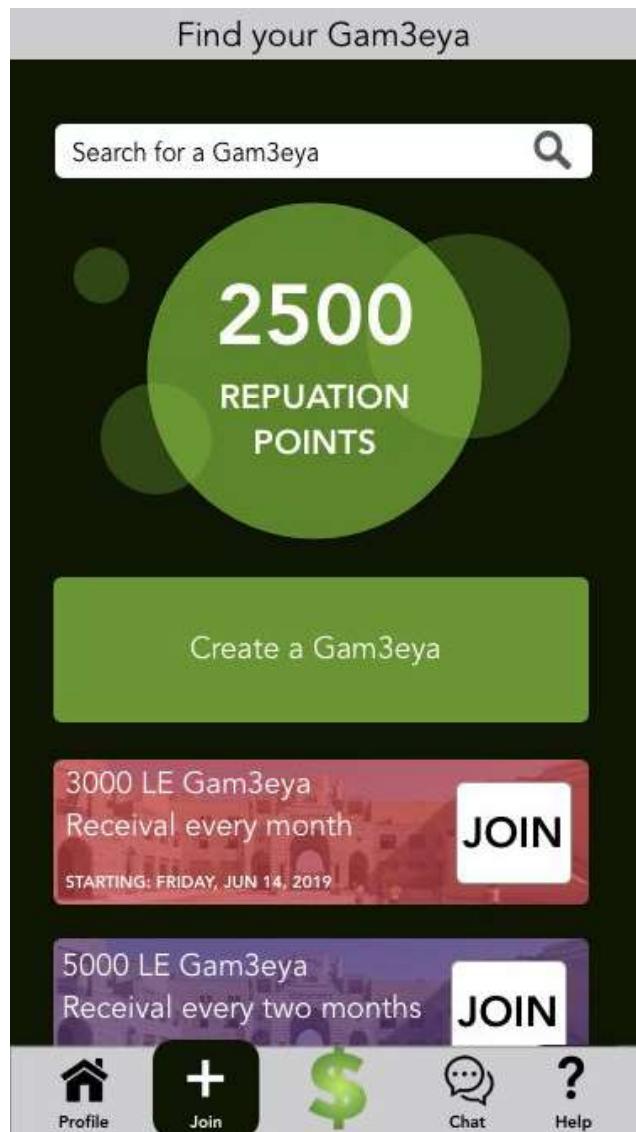
3. External Interface Requirements

3.1. User Interfaces

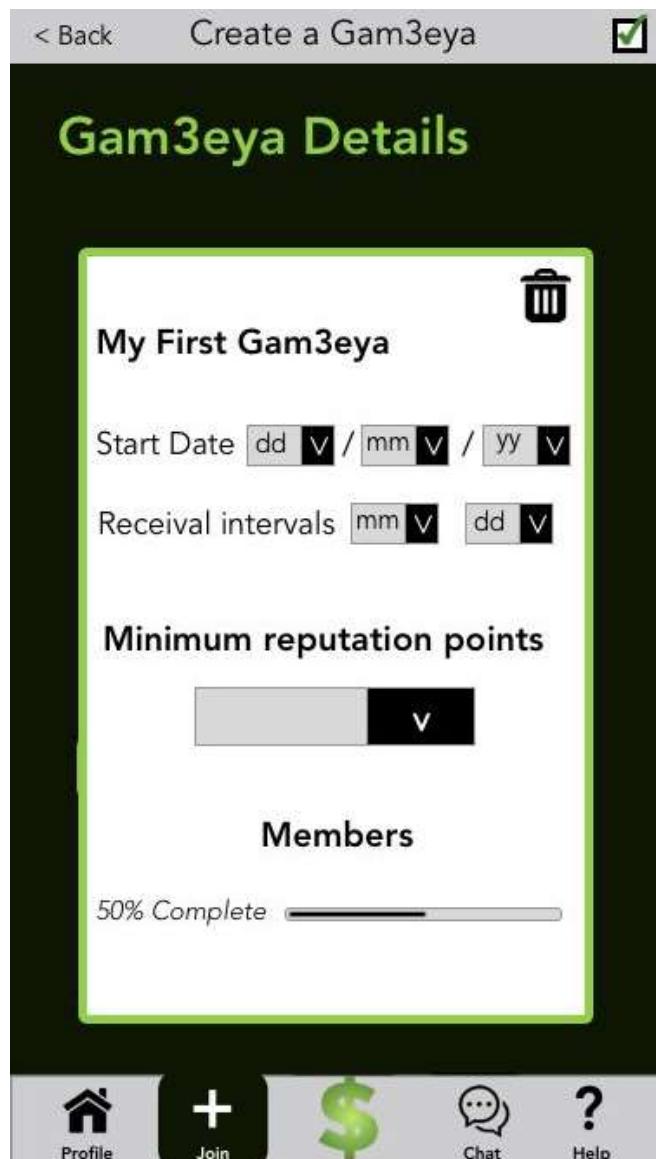
3.1.1 Login/Sign up Interface



3.1.2 Find a Gam3ya Interface



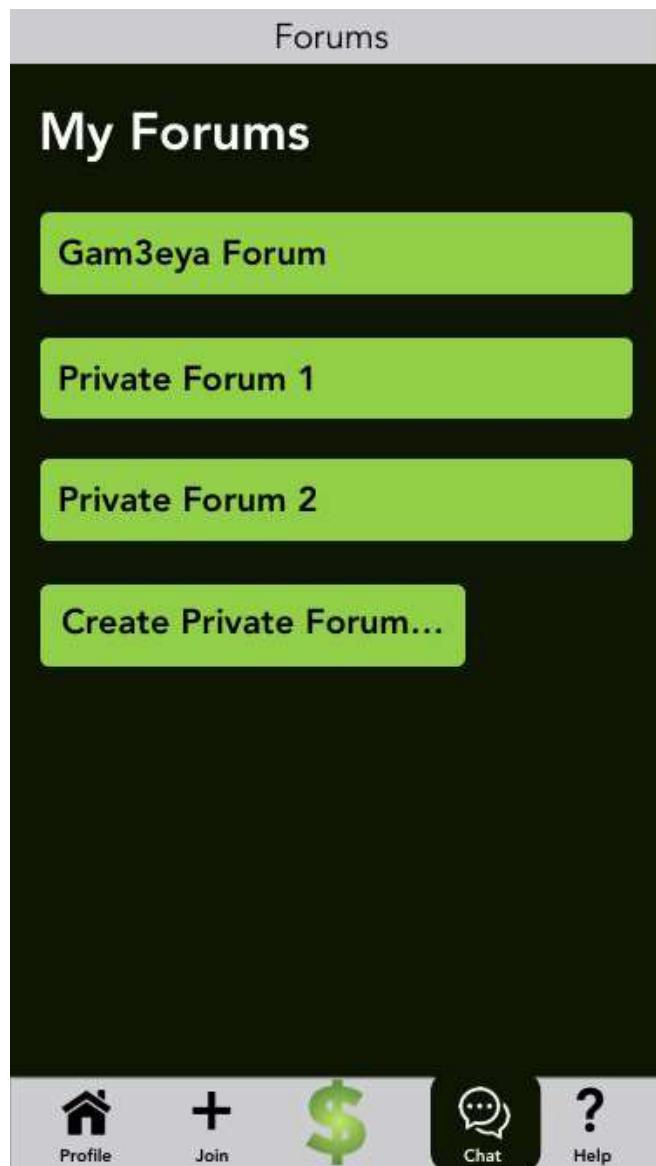
3.1.3 Create A Gam3ya Interface



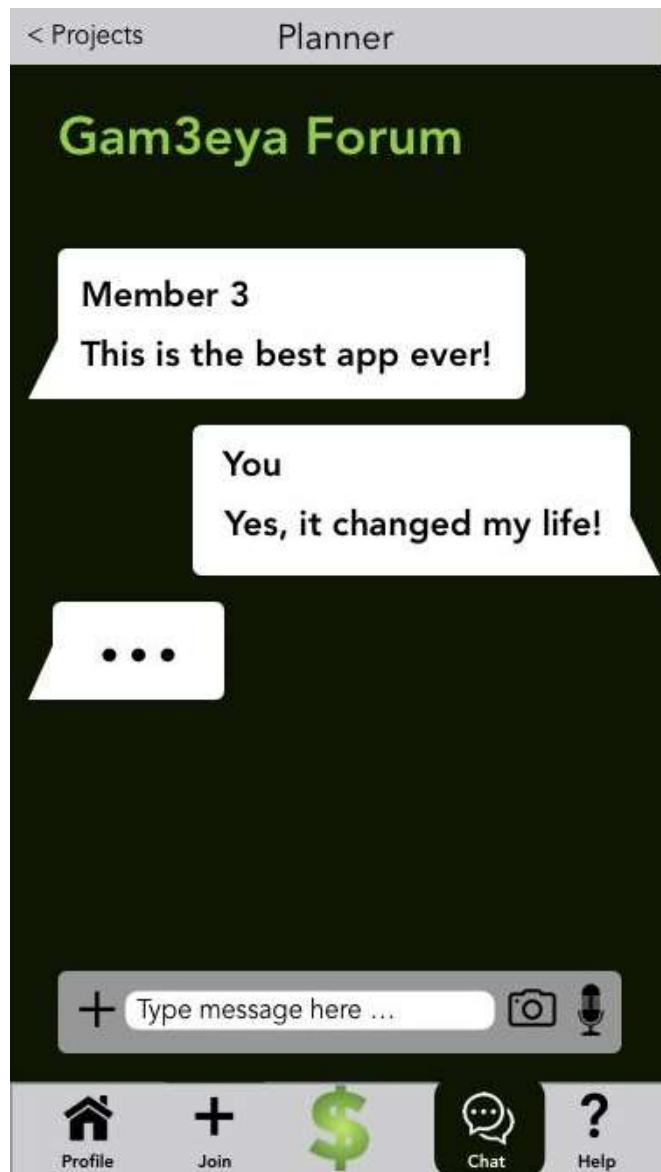
3.1.4 Gam3ya Details Interface



3.1.5 Forums Interface



3.1.6 Chat Interface



3.1.7 Help Interface



3.2. Hardware Interfaces

The potential hardware interfaces that Gam3ya might need are a server to host the chatting system, a server to host the Gam3ya's data and the devices through which the participants can pay and receive their payments. Nevertheless, all of these will be handled externally; we will be using cloud servers, specifically, Amazon Aurora, to host the chats and the data. Moreover, the payment interactions will be handled by Fawry, Vodafone Cash and Bitcoin Exchange services. Consequently, Gam3ya would not need to manage any hardware interfaces.

3.3. Software Interfaces

As mentioned in 3.2, we will not be using any hardware interfaces. As a result, all of our interaction will take place through software interfaces. To manage our data, we will be using Amazon Aurora API, which will grant us fast access to our data without the need to have a physical server. Regarding receiving and distributing funds, we will be using different interfaces to provide a wide range of options for the users. For the bitcoin, Coinbase will be used. Coinbase provides a convenient API through which receiving and distributing funds can be done in a matter of seconds. For local payments, we will use Fawry's API. Furthermore, Vodafone Cash and CIB Wallet (mobile applications) can also be used to provide a wider pool of options for the users.

3.4. Communications Interfaces

All of Gam3ya's services will be provided on an online platform. Furthermore, the API calls will include private financial records. Consequently, the aspect of security is crucial. To ensure data privacy, we will add HTTPS support to the API calls, instead of using the traditional HTTP protocol. Regarding the chatting services, we will use the XMPP to transfer messages in the forums.

4. System Features

4.1. Create an Account in Gam3ya

4.1.1 Description and Priority

To use any of Gam3ya's services, a user has to have an account. Once a user has an account, they will be able to join a Gam3ya, create a Gam3ya or join a chat forum. This feature is a high-priority feature.

4.1.2 Stimulus/Response Sequences

To create an account, the user has to press on the Create Account button on the home page. This will direct them to a form where they fill in the following:

- username (has to be unique)
- password (has to contain at least 8 characters including an Uppercase character, a lowercase character and a number)
- email address
- real name
- national ID number

Alternatively, users can sign up through their social media accounts such as Facebook or Twitter.

Use Case ID	1
Use Case Name	Create an Account
Description	Allows the user to create an account in Gam3ya
Actors	new Gam3ya users
Preconditions	The user is not logged in
Postconditions	A new account is created and added to the accounts' database.
Normal Flow	1.A new user visits the website 2.The user clicks on the Create an Account button 3.The user fills in the form 4.The Check National ID function is called and based on its response, an account is created or denied.

4.1.3 Functional Requirements

- AddAccountToDatabase()
- VerifyNationalID()
- createAccount()
- verifyAgeConstraint()

4.2. Check National ID

4.2.1 Description and Priority

To ensure that the system is not abused by bots, the user will have to provide the number of their national ID as well as their real name. This function takes these and uses external services that verify that this national ID actually exists. Moreover, the national ID will provide the user's age which can also be used to approve the creation of the new account

4.2.2 Stimulus/Response Sequences

This function is triggered whenever a user clicks on the Create Account button. It will receive the national ID and the real name and using third-party services, it will verify the user's identity

Use Case ID	2
Use Case Name	Check National ID
Description	It verifies the identity of the user
Actors	A governmental third-party service provider that verifies a national ID number along with the user's name
Preconditions	The user has filled the Create Account form correctly and clicked on the Create Account button
Postconditions	A response will be sent back to the Create Account function approving or rejecting the creation of the account
Normal Flow	1.A user clicks on the Create Account button after filling in the required form 2.A request is sent to the external National ID verification service 3.A response gets received and directed to the create account to either approve or reject the creation request.

4.2.3 Functional Requirements

- createAccount()
- verifyAgeConstraint()

4.3. Verify Age Constraint

4.3.1 Description and Priority

As explained in **4.10**, the national ID will be used to determine the users' age. Gam3ya is supposed to be for anyone who is 18 years or older. This function takes the national ID number and using it, it extracts the date of birth and calculates the age of the user.

4.3.2 Stimulus/Response Sequences

Once the national ID is checked, this function is invoked to calculate the user's age and return it to the Check National ID function to contribute in the approval or rejection decision.

Use Case ID	3
Use Case Name	Verify Age Constraint
Description	This function calculates the user's age to help decide whether they are fit to use the application or not
Actors	No actors. This function gets invoked internally once the Check National ID process starts
Preconditions	The user has filled the Create Account form correctly and clicked on the Create Account button
Postconditions	A response of whether the user's age qualifies them to have an account in Gam3ya or not is sent to the Create Account function to approve or reject the creation of the new account
Normal Flow	<ol style="list-style-type: none"> 1.The Create Account is Clicked after the data has been filled 2.The national ID number is used to extract the user's Birth date 3.The birth date is then used to calculate the user's age 4.A message is sent to the Create Account to allow or deny the user creating an account

4.3.3 Functional Requirements

- checkNationalId()
- verifyNationalID()
- createAccount()

4.4. Login to Gam3ya

4.4.1 Description and Priority

To use any of Gam3ya's services, a user has to have an account. Once a user has an account, they will be able to join a Gam3ya, create a Gam3ya or join a chat forum. This feature is a high-priority feature.

4.4.2 Stimulus/Response Sequences

To create an account, the user has to press on the Create Account button in the home page. This will direct them to a form where they fill in the following:

- username (has to be unique)
- password (has to contain at least 8 characters including an Uppercase character, a lowercase character and a number)
- email address
- real name
- national ID number

Alternatively, users can sign up through their social media accounts such as Facebook or Twitter.

Use Case ID	4
Use Case Name	Login to Gam3ya
Description	Allows the to login to Gam3ya
Actors	Any Gam3ya user that has created an account
Preconditions	The user must be logged out and must have created a Gam3ya account.
Postconditions	The user will be able to use all of the application's features (view a Gam3ya, search for a Gam3ya, join a Gam3ya, Create a Gam3ya etc...)
Normal Flow	<ol style="list-style-type: none"> 1.The user clicks on the Login Button 2.The user inputs their username and password 3.If the username and password were correct, the user gets access to Gam3ya's services.

Alternative Flow	1.The user clicks on the Login Button 2.The user inputs their username and password 3.The username and password are not corrected so the user gets denied access.
------------------	---

4.4.3 Functional Requirements

- AuthenticateLoginData()
- checkCredentialsInDatabase()
- grantAccess()

4.5. Use the tutorials

4.5.1 Description and Priority

This function allows the user to access video tutorials of how to use Gam3ya. This is a low priority function because users might still be able to operate without it.

4.5.2 Stimulus/Response Sequences

After logging in, a user can click on the Tutorials button and watch the videos to get a better understanding of how to use the system.

Use Case ID	5
Use Case Name	Use the tutorials
Description	This function allows users to preview videos of how to use the system
Actors	Gam3ya user
Preconditions	The user must have an account and have logged in to access the provided tutorials
Postconditions	One or more videos would have been watched by the user and hopefully they become better at using the system features by the end of these videos

Normal Flow	1.A user logs into Gam3ya 2.They click on the Tutorials button 3.They choose the first video and click on Start Video 4.They watch the video 5.They return to the home page to join/create Gam3ya
-------------	---

4.5.3 Functional Requirements

- accessTutorialDatabase()
- displayVideo()
- startTutorial()

4.6. Request support

4.6.1 Description and Priority

Through this function, a user can request support from a customer service agent to better understand the system and/or report a problem. This is a medium priority function: the system can function if it is down for a short period of time but it has to be fixed as soon as possible.

4.6.2 Stimulus/Response Sequences

A user clicks on Request Support button and a chat window with a customer service agent opens.

Use Case ID	6
Use Case Name	Request support
Description	This function allows users to press on a button to request support by a customer service agent
Actors	1.Gam3ya user 2.Customer Service Agent
Preconditions	The user must have an account and have logged in
Postconditions	A chatting window should be open where the user can chat with customer service agent

Normal Flow	1.A user logs in 2.They click on Request Support 3.A chatting window open 4.The user asks the questions they have 5.The customer service agent responds to the questions 6.The user ends the chat and returns to home page 7.The user displays Gam3yas to join one.
-------------	---

4.6.3 Functional Requirements

- sendMessage()
- receiveMessage()
- searchDatabase()

4.7. Search for a Gam3ya

4.7.1 Description and Priority

A user should be able to search for a Gam3ya that meets their needs. Nevertheless, only the Gam3ya's with a relatively close reputation points should be displayed.

This is a high priority function

4.7.2 Stimulus/Response Sequences

To search for a Gam3ya, a user will input the target monthly payment they are willing to pay and the system will display the available Gam3ya's

Use Case ID	7
Use Case Name	Search for a Gam3ya
Description	Allows the user to search for a Gam3ya that meets their needs
Actors	Gam3ya users
Preconditions	1.The user is logged-on 2.The user has input the target monthly payment they want to pay as well as the Gam3ya duration.

Postconditions	A list of Gam3yas shall be displayed along with the reputation of the average user in it. Moreover, the Gam3ya duration and monthly payments will be displayed as well.
Normal Flow	<ol style="list-style-type: none"> 1.The user inputs the Gam3ya duration and the monthly payment 2.The user clicks on the Search For a Gam3ya button 3.The user's reputation is checked and Gam3ya's with close average reputation that meet the user's needs are displayed.

4.7.3 Functional Requirements

- checkUserReputation()
- displayGam3eyas()
- searchDatabase()

4.8. Display Available Gam3yas

4.8.1 Description and Priority

To support the Search functionality, a function that displays the Gam3yas has to be utilized. This is a high priority function because if Gam3yas do not get displayed, users will not be able to join any Gam3ya.

4.8.2 Stimulus/Response Sequences

After the user inputs their needs and clicks on the Search button, a list of Gam3yas shall be displayed that meet their needs.

Use Case ID	8
Use Case Name	Display Available Gam3yas
Description	To support the Search functionality, a function that displays the Gam3yas has to be utilized.
Actors	Gam3ya users
Preconditions	<ol style="list-style-type: none"> 1.The user is logged in 2.The user has input their needs and clicked on the Search button
Postconditions	A list of Gam3yas will be displayed that meet the user's needs

Normal Flow	1.The search Button is clicked 2.The user's reputation gets checked 3.The database of Gam3ya's is searched and the Gam3ya's that meet the user's need and are of close average reputation are displayed
-------------	---

4.8.3 Functional Requirements

- `searchDatabase()`
- `displayGam3ya()`
- `checkUserReputation()`

4.9. Join a Gam3ya

4.9.1 Description and Priority

This is one of the core functions in the system and carries a high priority. It allows the user to submit a request to join a specific Gam3ya. If the request is approved, the user will join the chosen Gam3ya and will be required to pay the pre decided amount of money every month.

4.9.2 Stimulus/Response Sequences

To join a Gam3ya the user will submit a request via this function and the request would be processed. When the request is submitted, another function is triggered: GetMembersApproval which collects the data after the current members in the Gam3ya vote to permit or reject the request. Based on the votes the user's request would either be granted or refused.

Use Case ID	9
Use Case Name	Join Gam3ya
Description	Allows the user to submit a request to join a Gam3ya and if it is granted, they would join their chosen Gam3ya
Actors	Gam3ya users
Preconditions	The user is logged in and have searched for a Gam3ya
Postconditions	A user would be admitted to a Gam3ya and they would automatically join the Gam3ya's forum if their request gets approved by the other members; otherwise, the user is informed that their request has been disapproved and thus, they cannot join their chosen Gam3ya

Normal Flow	<ol style="list-style-type: none"> 1.A user with high reputation points (a lot more than the required amount to join the Gam3ya) searches for a Gam3ya with specific preferences 2.Available Gam3ya for them are displayed (based on reputation points) 3.The user selects a Ga3mya and submits a request to join 4.Their request gets processed by the other members of the Gam3ya 5.The user's reputation points are checked and the majority of the members in the Gam3ya approve the request 6.The user is admitted to that Gam3ya 7.The user joins the Gam3ya's forum
Alternate Flow	<ol style="list-style-type: none"> 1.A user with average reputation points (barely exceeds the minimum amount required to join the Gam3ya) searches for a Gam3ya with specific preferences 2.Available Gam3ya for them are displayed (based on their reputation points) 3.The user selects a Ga3mya and submits a request to join 4.Their request gets processed by the other members of the Gam3ya 5.The user's reputation points are checked and the majority of the members in the Gam3ya disapprove the request 6.The user is informed that their request has been disapproved and thus, they cannot join their chosen Gam3ya

4.9.3 Functional Requirements

- SearchForAGam3ya()
- ApproveMembershipRequest()
- joinGam3yaForum

4.10. Create a Gam3ya

4.10.1 Description and Priority

This function has a high priority and it is used by users to create customized Gam3ya's. It allows the user to submit a request to create a Gam3ya with custom monthly payments and a number of participants. If the request is approved, the user will be granted permission and thus, the chosen Gam3ya will be created.

4.10.2 Stimulus/Response Sequences

To create a Gam3ya the user will submit a request via this function and the request would be processed. When the request is submitted, another function is triggered: CheckReputation which checks the reputation points of a given user. Based on the reputation points the user's request would either be granted or refused.

Use Case ID	10
Use Case Name	Create a Gam3ya
Description	Allows the user to submit a request to create a Gam3ya and if it is granted, the Gam3ya would be created and they would be the first member to be added to the Gam3ya.
Actors	New Gam3ya users
Preconditions	The user is logged in.
Postconditions	A user would be granted permission to create a Gam3ya if their request gets approved based on their reputation points, the Gam3ya's monthly payments, etc... Then, a forum for the Gam3ya would be automatically created and the user would be added to it. If the request was not approved, the user would be informed that their request has been disapproved and thus, they cannot create a Gam3ya with the chosen preferences given their reputation points.
Normal Flow	<ol style="list-style-type: none"> 1.A user with high reputation points (more than the required amount to create a Gam3ya) attempts to create a Gam3ya with specific monthly payments and number of members 2.Based on their reputation points (which are relatively high) their request gets approved 3.The Gam3ya is created and the user is added to it and it is now available to be searched for by other members. 4.A forum for the Gam3ya is created

Alternate Flow	<p>1.A user with low reputation points (less than the required amount to create a Gam3ya) attempts to create a Gam3ya with a high monthly payment.</p> <p>2.Based on their reputation points (which are relatively low) their request gets rejected</p> <p>3.The user is informed that they are not given permission to create a Gam3ya with that monthly amount due to their reputation points.</p>
----------------	--

4.10.3 Functional Requirements

- createGam3yaForum()
- checkUserReputation()
- givePermissionToCreateGam3ya()

4.11. Check Reputation Points

4.11.1 Description and Priority

This function checks a user's reputation. This is a medium priority function because failure to check the reputation would cause all the Gam3ya's to be displayed to the user instead of the ones that are close to them in terms of reputation. This will not cause any fatal damage but it will make it more difficult for the user to get their request to join a Gam3ya approved.

4.11.2 Stimulus/Response Sequences

A user's reputation is requested so it gets checked and sent to the module that requires it.

Use Case ID	11
Use Case Name	Check Reputation Points
Description	This function is responsible for searching through the database for a specific user to check their reputation and deliver it to the subsystems that need it.
Actors	No actors
Preconditions	One of two things must happen before this function is called: 1.A user requests to create a Gam3ya 2.A user tries to search for Gam3yas
Postconditions	The reputation of a user is delivered to the module that needs it

Normal Flow	1.A user clicks on Create Gam3ya button 2.The user's reputation gets checked from the database 3.A response is sent to the creation request to either allow or deny it.
-------------	---

4.11.3 Functional Requirements

- getReputationFromDatabase()
- checkPaymentLog()

4.12. Give Permission To Create a Gam3ya

4.12.1 Description and Priority

This function assesses the user's reputation and if it exceeds the threshold, they are allowed to create a Gam3ya. Otherwise, their request gets denied.

4.12.2 Stimulus/Response Sequences

This function is triggered whenever a user sends a request to create a Gam3ya. The reputation is checked and a decision is made

Use Case ID	12
Use Case Name	Give Permission To Create A Gam3ya
Description	This function decides whether a user can create a Gam3ya or not.
Actors	No actors
Preconditions	A user clicks on the Create Gam3ya Button
Postconditions	A decision is made and sent to the user of whether the request is approved or denied.
Normal Flow	1.A user clicks on the Create Gam3ya Button 2.The reputation of the user is checked 3.Based on the reputation, the user either gets allowed to create the Gam3ya or the request gets denied.

4.12.3 Functional Requirements

- checkReputation()
- checkPaymentLog()

4.13. Create a Gam3ya Forum

4.13.1 Description and Priority

This function has a medium priority and it is used to create a forum for a Gam3ya. It is not directly accessed by the user; rather it is automatically created whenever a new Gam3ya is created. The forum could be used to allow text messages or video calls between members of the Gam3ya.

4.13.2 Stimulus/Response Sequences

When a Gam3ya is created, this function is triggered which creates a forum; whenever a new member joins the Gam3ya, they are directly added to this forum

Use Case ID	13
Use Case Name	Create a Gam3ya Forum
Description	A forum is automatically created when a new Gam3ya is created
Actors	This function is not associated with actors; rather, it is automatically triggered when a new Gam3ya is created.
Preconditions	A new Gam3ya is created
Postconditions	A forum is created and any member that joins this Gam3ya would be automatically added to it.
Normal Flow	<ol style="list-style-type: none"> 1.A user creates a Gam3ya (described in the normal flow of the Create a Gam3ya usecase) 2.The Gam3ya is created 3.The forum gets automatically created

4.13.3 Functional Requirements

- createGam3ya()
- createForum()
- checkReputationPoints()

4.14. Create a Private Forum

4.14.1 Description and Priority

This function has a medium priority and it is used to create a private forum. A private forum is created whenever a user requests to create and host one. Members of the software (not necessarily from the same Gam3ya) would be allowed to join this private forum upon the approval of the forum creator (moderator).

4.14.2 Stimulus/Response Sequences

When a user decides to create a private forum, this function is triggered which creates a private forum.

Use Case ID	14
Use Case Name	Create a Private Forum
Description	A forum is automatically created when a new Gam3ya is created
Actors	Forum Creator
Preconditions	The user needs to be logged in
Postconditions	A private forum is created and the creator would become the moderator that accepts or rejects requests to join the private forum
Normal Flow	<ol style="list-style-type: none"> 1.A user logs in 2.They create a private forum 3.User becomes the moderator of the private forum 4.Two people apply to join the private forum 5.The moderator allows one and rejects the other (just an example)

4.14.3 Functional Requirements

- createPrivateForum()
- joinPrivateForum()
- ApproveJoinForumRequest()
- RejectJoinForumRequest()

4.15. Join a Private Forum

4.15.1 Description and Priority

This function has a medium priority and it is used to allow members to send requests to join a private forum. All members of the software can send requests to join any private forum.

4.15.2 Stimulus/Response Sequences

When a user decides to join a private forum, this function is triggered which sends a request to the forum moderator; if the moderator approves the request, the user is given permission to join the private forum where they can socially interact with all members of the forum via text messages, video and phone calls. If the moderator rejects the request, the user wouldn't be allowed to view the interactions taking place within the forum.

Use Case ID	15
Use Case Name	Join a Private Forum
Description	This function allows users of the system to send requests to join private forums to socially interact with the members of the private forums.
Actors	Gam3ya User, Forum Creator
Preconditions	The user needs to be logged in
Postconditions	If the request is approved, the user would be granted access to the forum and would have the ability to view all the interactions taking place. Otherwise, the user would be informed that their request has been rejected.
Normal Flow	<ol style="list-style-type: none"> 1.A user logs in 2.They send a request to join a private forum 3.The moderator approves the request 4.The user starts to interact with the moderator via text messages

4.15.3 Functional Requirements

- joinPrivateForum()
- ApproveJoinForumRequest()
- RejectJoinForumRequest()
- SendTextMessage()
- UseVideoCall()
- UseVoiceCall()

4.16. Join a Gam3ya Forum

4.16.1 Description and Priority

This function has a medium priority and it is used to allow members to join a Gam3ya forum. In this sort of forums, only members of a specific Gam3ya would be granted access to this forum.

4.16.2 Stimulus/Response Sequences

When a user joins a specific Gam3ya, this function is automatically triggered which adds the user to the forum automatically. All the users of the forum would be able to socially interact with one another. The first member that joins a Gam3ya (and consequently, the first one to join the Gam3ya forum) becomes the moderator, having the ability to kick and report other members for the misuse of the forum.

Use Case ID	16
Use Case Name	Join a Gam3ya Forum
Description	This function allows users of the system to send requests to join private forums to socially interact with the members of the private forums.
Actors	This function has no direct association with actors. Although it allows the users to join the Gam3ya forum, it is solely triggered by the JoinGam3ya() function independently from users.
Preconditions	The user joins a Gam3ya
Postconditions	The user is added to the Gam3ya forum
Normal Flow	1.A user logs in 2.A user joins a Gam3ya (described in the normal flow of the Join a Gam3ya usecase) 3.The user gains access to the Gam3ya Forum 4.The user starts to interact with the members of the forum

4.16.3 Functional Requirements

- joinGam3yaForum()
- SendTextMessage()
- UseVideoCall()
- UseVoiceCall()

4.17. Get Members' Approval

4.17.1 Description and Priority

This function has a medium priority and it is used to gather the votes of the members of a Gam3ya in order to decide whether a new member should be admitted to the Gam3ya or rejected.

4.17.2 Stimulus/Response Sequences

When a user sends a request to join a specific Gam3ya, this function is automatically triggered which asks all the current members of the Gam3ya for their votes regarding the new member. The members of the Gam3ya can view the new member's reputation and they can vote to accept or reject his request. Based on the majority of the vote, the decision is taken.

Use Case ID	17
Use Case Name	Get Members' Approval
Description	This function is used to gather the votes of members within a Gam3ya regarding the admission or rejection of a new member to the Gam3ya
Actors	Gam3ya Users
Preconditions	The user sends a request to join a Gam3ya
Postconditions	The user is either added to the Gam3ya or their request is rejected
Normal Flow	<ol style="list-style-type: none"> 1.A user logs in 2.A user sends a request to join a Gam3ya 3.The members are asked to vote whether to allow or disallow the newest member from joining 4.Check the votes 5.Allow or reject the member from joining the Gam3ya

4.17.3 Functional Requirements

- joinGam3ya()
- collectMembersVotes()
- joinForum()

4.18. Complete a Gam3ya

4.18.1 Description and Priority

This function has a medium to low priority and it includes rewarding the users at the end of every Gam3ya for the commitment to the Gam3ya.

4.18.2 Stimulus/Response Sequences

This function triggered once the user pays the last monthly payment of a Gam3ya and the Gam3ya is considered complete; the user's reputation points that are used for assessment are increased as a result of their dedication to the Gam3ya for its whole period.

Use Case ID	18
Use Case Name	Complete a Gam3ya
Description	This is a function that is called at the end of the Gam3ya in order to reward those dedicated members of the Gam3ya who abided by the Gam3ya's rules.
Actors	Gam3ya User
Preconditions	A user pays the last monthly payment of the Gam3ya
Postconditions	Trigger the function to increase the user's reputation points
Normal Flow	<ol style="list-style-type: none"> 1.A user joins a Gam3ya (described in Join a Gam3ya) 2.The user pays their monthly payments regularly for the Gam3ya's period 3.The Gam3ya finishes 4.The function to reward the user is triggered

4.18.3 Functional Requirements

- payMonthlyPayment()
- increaseReputationPoints()

4.19. Increase Reputation Points

4.19.1 Description and Priority

This function is the core of the assessment system in the product. It is the system's way of rewarding the users that are loyal to the Gam3ya until it finishes.

4.19.2 Stimulus/Response Sequences

This function triggered automatically once a Gam3ya finishes. All members of the Gam3ya would be rewarded and their reputation points would be increased.

Use Case ID	19
Use Case Name	Increase Reputation Points
Description	This function is responsible for adding some reputation points to the users when a Gam3ya, to which they are members, finishes.
Actors	Gam3ya User
Preconditions	A Gam3ya is finished
Postconditions	The user would have an increased amount of reputation points
Normal Flow	<ol style="list-style-type: none"> 1.A user joins a Gam3ya (described in Join a Gam3ya) 2.The user pays their monthly payments regularly for the Gam3ya's period 3.The Gam3ya finishes 4.This function is triggered and as a result, the user is rewarded and their reputation points increase

4.19.3 Functional Requirements

- payMonthlyPayment()
- increaseReputationPoints()

4.20. Reduce Reputation Points

4.20.1 Description and Priority

This function is, along with Increase Reputation Points, represent the core of the assessment system that is being used. It is the system's way of penalizing the users that don't abide by the deadlines for their payments.

4.20.2 Stimulus/Response Sequences

This function triggered automatically once a user misses the deadline to pay their monthly payment. Depending on the number of days the payment is late, the amount deducted from the user's reputation points increases.

Use Case ID	20
Use Case Name	Reduce Reputation Points
Description	This function is responsible for deducting some reputation points from the users that don't pay the pre agreed upon amount of time.
Actors	Gam3ya User
Preconditions	A user's payment deadline has passed
Postconditions	Decrease the user's reputation points
Normal Flow	<ol style="list-style-type: none"> 1.A user joins a Gam3ya (described in Join a Gam3ya) 2.The user does not pay their monthly payments on time 3.This function is triggered and as a result, the user is penalized and their reputation points are decreased

4.20.3 Functional Requirements

- payTheMonthlyAmount()
- reduceReputationPoints()

4.21. Send Message

4.21.1 Description and Priority

This function has a medium to low priority and it includes sending a message from one user to other user(s) in a forum. The message can be text, audio or video.

4.21.2 Stimulus/Response Sequences

A user can click on any of the forums they joined and start typing or sending audio/video.

Use Case ID	21
Use Case Name	Send Message
Description	This is a function called that takes a message from a user and delivers it to other users of the same forum.
Actors	Gam3ya User
Preconditions	The user must be logged in and he must have already joined a forum.
Postconditions	One or more other users receive a message.

Normal Flow	1.The user logs into his/her account 2.The user opens one of the forums they joined. 3.The user sends a message. 4.The message is forwarded to all members of the forum.
-------------	---

4.21.3 Functional Requirements

- sendTextMessage()
- sendAudioMessage()
- sendVideoMessage()

4.22. ReceiveMessage

4.22.1 Description and Priority

A user receives a message and it gets displayed in the forum

4.22.2 Stimulus/Response Sequences

When a user sends a message, other user(s) are supposed to receive this message.

Use Case ID	22
Use Case Name	Receive Message
Description	A user receives a message and it gets displayed in the forum
Actors	Gam3ya User
Preconditions	1.The user must be logged on 2.Another user must have sent a message.
Postconditions	The message is displayed in the forum where it can be seen by all the participants in that forum

Normal Flow	1.A user sends a message. 2.The message gets displayed in the forum in which it was sent. 3.A user in the forum views the sent message and responds by another message.
-------------	---

4.22.3 Functional Requirements

- receiveTextMessage()
- receiveAudioMessage()
- receiveVideoMessage()

4.23. Pay the monthly amount

4.23.1 Description and Priority

This function has a high priority. This function allows the financial service provider to update a user's account with the payment they made.

4.23.2 Stimulus/Response Sequences

Use Case ID	23
Use Case Name	Pay the monthly amount
Description	This function adds the amount of money paid to the account of the Gam3ya user.
Actors	Financial Service provider, Gam3ya User
Preconditions	The user must be in a Gam3ya and he must have paid the money to the financial services provider.
Postconditions	The user's account is incremented with the amount of money paid.
Normal Flow	1.A Gam3ya user pays the amount due in person to the financial services provider before the deadline. 2. The financial services provider updates a user account with the amount paid. 3.One of the Gam3ya members receives the money collected from all members.

Alternative Flow	<ol style="list-style-type: none">1. A Gam3ya user pays the amount due in person to the financial services provider after the deadline.2. The financial services provider updates the user's account with the amount due.3. Reputation points are deducted from that user's account as per section 4.163. One of the Gam3ya members receives the money collected from all members.
-------------------------	---

4.23.3 Functional Requirements

- payMonthlyPayment()
- increaseReputationPoints()
- payThroughFawry()
- payThroughCoinbase()
- payThroughVodafoneCash()
- payThroughCIBWallet()

4.24. Receive payment

4.24.1 Description and Priority

This function informs the financial services provider that it is time for a specific Gam3ya user to receive their payment.

4.24.2 Stimulus/Response Sequences

This function decreases the balance of the Gam3ya's user with the amount that he/she is going to receive and allows the financial services provider to disburse the amount of money agreed upon.

Use Case ID	24
Use Case Name	Receive payment
Description	This function deducts the amount of money paid to the account from the Gam3ya user.
Actors	Financial Service provider, Gam3ya User
Preconditions	The user must be in a Gam3ya and at least one of the other users must have paid their amounts.
Postconditions	The user's account is decremented with the amount of money received.
Normal Flow	<p>1. The user must be enrolled in a Gamya.</p> <p>2. All of the other Gam3ya users pay their due amounts before their deadlines.</p> <p>3. The financial services provider is notified that that Gam3ya that a particular user has the right to receive the full amount of money specified.</p> <p>4. The amount of money received is deducted from the user's account.</p>
Alternative Flow	<p>1. The user must be enrolled in a Gamya.</p> <p>2. Only some of the other Gam3ya users pay their due amounts before their deadlines and others don't make their payments.</p> <p>3. The financial services provider is notified that that Gam3ya that a particular user has the right to receive a part of the amount of money specified.</p> <p>4. The amount of money received is deducted from the user's account.</p> <p>5. When the rest of the Gam3ya members pay their due amounts, the user is notified and the financial services provider is notified.</p> <p>6. The amount received is deducted from the user's account.</p>

4.24.3 Functional Requirements

- receiveMonthlyPayment()
- notifyUserWithNewAmount()
- receivePaymentThroughFawry()
- receivePaymentThroughCoinbase()
- receivePaymentThroughVodafoneCash()
- receivePaymentThroughCIBWallet()

4.25. Manage the funds

4.25.1 Description and Priority

This function is responsible for monitoring the flow of money from the users to Gam3ya's bank account. It is also a high priority function

4.25.2 Stimulus/Response Sequences

Use Case ID	25
Use Case Name	Manage the funds
Description	This function allows a finance agent to view the transactions made by the users in order to be able to know if something went wrong with financial transactions.
Actors	Bank, Finance Agent and Financial Service Providers
Preconditions	There must be at least one transaction made on the system and there must be a complaint from the user or a self-generated error about a financial transaction that was not recorded into the system or a mistake in a financial record.
Postconditions	The financial records of one or more users could be edited.
Normal Flow	<ol style="list-style-type: none"> 1.A user files a complaint that there is an error in their financial records or in a specific financial transaction. 2.The bank, the finance agent and the financial service provider review their complaints. 3.The bank, the finance agent and the financial service provider modify the financial records for that user to fix the reported issue. 4.A record must be made of the modification made. 5.The user is informed if there is a modification in their financial records.

4.25.3 Functional Requirements

- checkUserPayment()
- checkBankAccount()
- chechPaymentsArrival()
- manageFunds()

5. Other Nonfunctional Requirements

5.1. Performance Requirements

The target users of the Gam3ya system can be estimated to be around 6000 users. A server will be used to host the database, application and the website. The server needs to work 24/7 with a minimum possibility of failure. In the case of a failure of the server, financial transactions may not get recorded on the system. This will lead to the possibility of clients claiming that they made their payments, whereas the system will tell them that they did not pay their installments. For that reason, failure of the server must be highly avoided. That is why Amazon Web Services are going to be used to host the application, website and the database. Specifically, Amazon Aurora is the service that is going to be used. Amazon Aurora offers the opportunity of having virtual machines on the servers. Those virtual machines will be used for data analytics of the data of the users. It is estimated that Amazon Aurora will cost around 80 US \$ per month. There are other requirements, such as the speed of the network at the user's end, however, this should be the responsibility of the user and the user must be informed about that in the terms and conditions of the agreement. In addition, the requirement of non-failure of payment services, such as Fawry and Vodafone Cash have to be handled by the third-party corporations and are not the responsibility of our firm.

5.2. Safety Requirements

There are little or no hardware components in the system; therefore, there are no major concerns about the physical safety of the users of the system.

5.3. Security Requirements

Security has to be considered from many perspectives: the privacy of the data, the security of the data against cyber attacks. Concerning the privacy of data, access rights to the data have to be defined. The users of the Gam3ya will be able to see only the username and the reputation points of other users of Gam3ya. The administrators of the systems will be able to view all the details of Gam3ya members (including financial information), however, they will not be able to modify any of this data. Fawry or any other financial services provider will be able to view the balance of a user and their username in order to send receipts with their usernames whenever a user makes a payment. The security of the data against any malicious attack must be guaranteed. There are two aspects of guaranteeing the security of the data. One of them is having an antivirus on the servers and that will be handled by Amazon. The other is guaranteeing that there is no malicious software on the user's mobile phone, which the firm will have no control over other than not allowing cookies to access the metadata of the application stored on the device.

5.4. Software Quality Attributes

The quality of the software will be encapsulated in the availability, reliability and security. The software will be available in Egypt, since the application will be linked only to Egyptian financial service providers, such as Fawry and Vodafone Cash, so the application will have to be modified if

the firm intends to use it outside of Egypt. Addressing portability, the application will have to be implemented as an application to be used on both Android and IOS as well as a web application to be used on web browsers. After agreeing on the terms and conditions of using the application, there will be a tutorial to help users understand the functionality of the application. Also, Gam3ya offers a customer support service for its users to increase user satisfaction level. Another factor to be considered is maintainability. There is no need to worry about the maintenance of the databases, the servers hosting the application, or the payment methods/services since all of these will be handled by third party organizations.

5.5. Business Rules

The Gam3ya company will be a for-profit company. The profit in Gam3ya will be generated only from two sources, the commission the users pay whenever they receive their payment and the interest generated from saving money from low reputation Gam3ya's in a bank account. Apart from the profit, using advertisements on the website will not be allowed in order to make using the application more convenient for the users. Also, selling the results of data analytics on the data of customers will not be allowed to protect the privacy of the users. If there is a delay in a payment by a user, there would be no legal action taken against that user. That being said, a penalty would be issued which reduces their reputation. In the case of saving the capital money of low reputation Gam3ya's in a bank, the interest of the bank will not be distributed among the users of Gam3ya; rather, it would be used to compensate for users who refuse to continue paying after they receive their payments. Although this wouldn't eliminate the loss, it would reduce the total loss amount. Furthermore, The financial service providers, such as Fawry will receive their commission as a percentage of the payment of the user, and such information would be agreed upon by the user in the terms and conditions of the application. It is expected that the target users of Gam3ya will not be high income citizens and therefore, they are less likely to own a credit card or have bank accounts. As a result, a user's account at Gam3ya will not be linked to bank accounts.

6. Other Requirements

One major requirement that was not covered elsewhere is the legal requirement of using Bitcoins. As far as we understand, trading in Bitcoins is illegal in Egypt. Nevertheless, the client has explicitly stated that they have obtained a license to allow trading in Bitcoins. Consequently, the inclusion of the feature of paying and receiving funds through bitcoins is the customer's liability.

Appendix A: Glossary

The key terms used in this SRS document are outlined below with detailed explanation of what exactly is meant by each term; this is used to improve the readability of this document and to help reduce confusion.

API: It is an acronym for Application Programming Interface. In essence, it is a simple communication mechanism that allows a client to communicate with the server (service provider) through an agreed upon message format.

HTTPS: Hypertext Transfer Protocol Secure is a secure protocol that allows the transfer of data over the internet.

XMPP: Extensible Messaging and Presence Protocol (XMPP) is a communication protocol used in messaging services to allow users to send and receive messages from each other in a quick manner.

Gam3ya: A Gam3ya refers to simply the collection of monthly payments from different members and the distribution of the total amount to a defined number of members (usually one) every month. The Gam3ya runs for a fixed and predefined number of months.

Gam3ya Participant: A member in a gam3ya is a person who agrees to pay a fixed amount of money every month and is to receive a payment of a total equal to the period of the Gam3ya (in months) multiplied by the pre decided monthly amount that is to be paid.

Low-reputation user: A user that is either still new to the system, or they have not been paying their payments on time.

Low-reputation Gam3ya: A gam3ya in which the average user reputation is low

High-reputation user: A user that has been a user for a long time and they have been paying their payments on time.

High-reputation Gam3ya: A gam3ya in which the average user reputation is high

Appendix B: The proposed solution to the client's concerns

There would be reputation points for each user that would be used for various reasons:

- The Gam3ya's limit would be based on the members' reputations
- Reputation points would help assess users based on whether they pay on time or not
- The **type** of Gam3ya and the when the first payment is distributed (explained below)

In order for the system to have a better financial security we plan to have many types of Gam3yas

- Low reputation (safe Gam3ya) :The payments distributed every month would be to more than 1 user but the first payment would be after a predefined number of months (for example starting from the 7th month, 2 users would get their payments instead of having 1 user get their payment from the first month). The money obtained would be invested (in a bank possibly) and the generated interest would help reduce the risks of failure (a full mathematical model would be provided to reinforce this point). The number of months before the first payment is distributed is likely to vary based on the reputation of the Gam3ya's users and the amount of money to be distributed.
- Regular Gam3ya: Only users with high reputation would be able to join such Gam3ya's. This Gam3ya would operate in the traditional fashion. Payments would be distributed from the first month.

Participants would submit their preferences and when they ideally want to get paid:

- In order to achieve a high level of fairness, users would get to choose when they want to be paid. The data of all users will then be combined and random combinations satisfying their preferences shall be generated. Moreover, to incentivise people to choose to get paid at the end, vouchers/discounts can be distributed to them as a form of compensation.

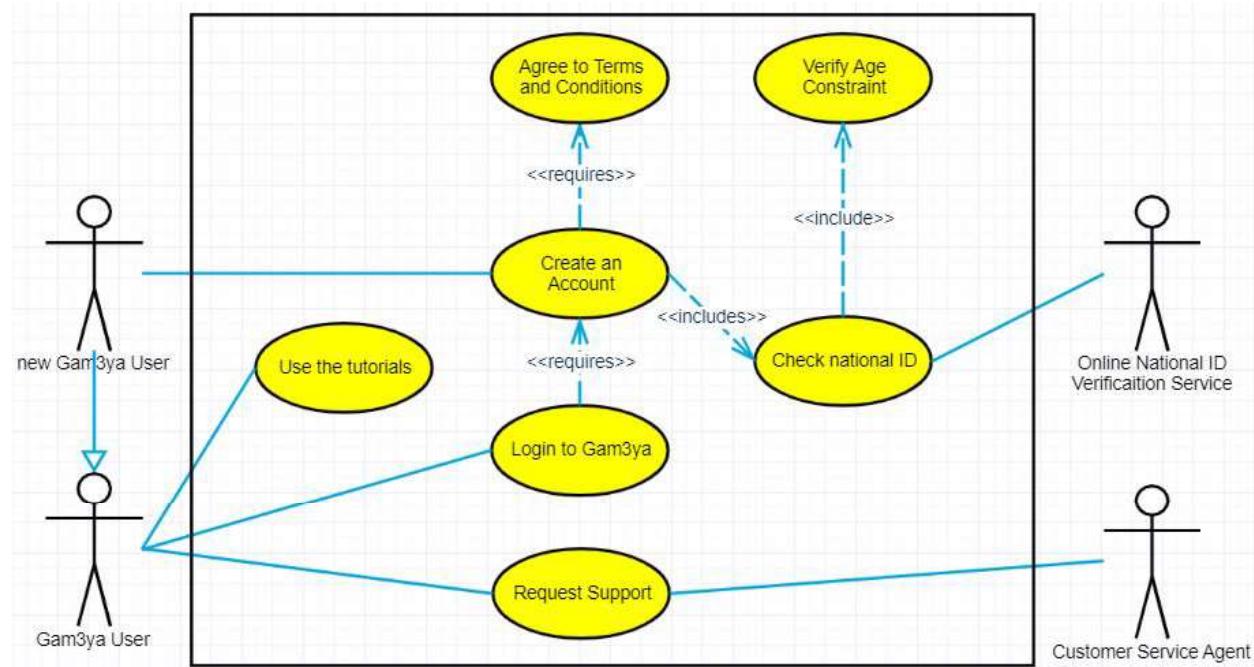
Appendix C: The client's interview questions

1. The description states that you are seeking to enable peer to peer lending. Can you please specify who can initiate a Gam3ya? Is it the user or the software?
 - a. Is there a minimum number of participants in a Gam3ya?
2. Can the payment policy be tweaked a little bit for various reasons including ensuring financial security? (Please look at Appendix B)
3. Gam3ya should be accessed on-line from a variety of devices. Can you please elaborate on who is our target user?
 - a. This allows us to decide the platform (mobile, web, etc...) that we will be using?
4. According to our research, dealings in cryptocurrencies are not fully legalized in Egypt. As our contractor, have you obtained a license to legally allow trading in cryptocurrencies?
5. What currencies should Gam3ya support?
6. What do you think of adding penalties on users who don't pay on time?
7. What exactly do you mean by "Mobile Operators"?
8. Would you like to take legal action if participants fail to pay?
 - a. What documents would we need to hold the participants accountable from a legal perspective?
9. What do you mean by "Security of Information"? Payment History? Date of Birth? Chat history? All?
10. We are planning to let users choose their preferred month in which they get paid to help construct the Gam3yas. Do you think that would be a viable solution?
11. What policies would you like to implement to ensure fairness? (Please look at Appendix B to be familiar with our plan)
12. Gam3ya should be used for social interactions. Can you list all the required functionalities that, in your opinion, would allow social interactions (text messages, video chatting, calls, etc...)? Should interactions be anonymous or should users be able to know the identities of others?
13. Would you like chatting to be possible between people from the same Gam3ya only or through participants' unique ID?
14. Is Gam3ya going to be operating in the public or the private sector? If it's in the private sector, would there be any association with the government and/or banks? (could be used to better predict the users' ability to repay the money they owe)
 - a. Banks can provide high-interest account where we can invest the money we get from the users.
15. Is Gam3ya aiming to generate profits? If yes, what are the proposed main sources of revenue? (could there be a subscription fee or a commission from each Gam3eya?)
 - a. If not, how else will you sustain the business?
16. What would you propose to minimize the loss when people fail to pay?
17. Would users be able to choose the Gam3ya they join (possibly with friends and colleagues) or choosing the members in each Gam3ya would be managed by the software?
18. Should all users of the same Gam3ya Pay the same amount every month?

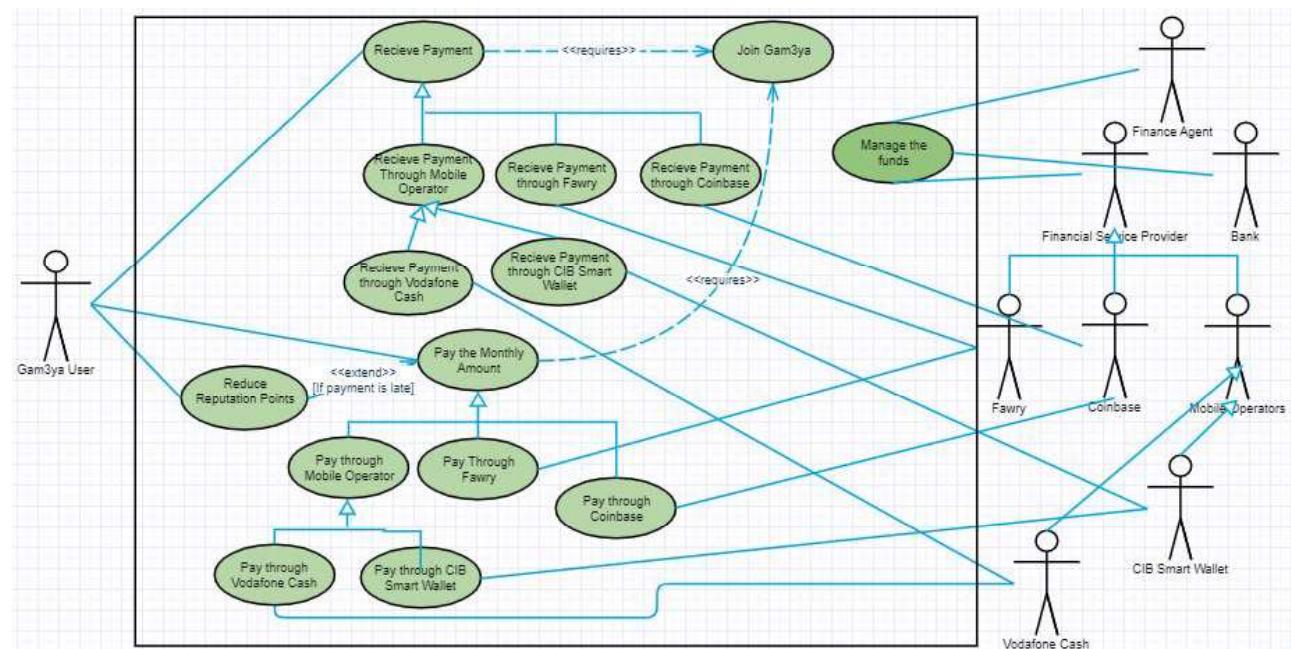
19. Customers' need to feel safe about using our application. What do you propose to solve this issue?
20. Can we allow the users to pre-pay and keep a record of it in their account?
21. Can users participate in more than one gam3ya?
22. Will we consider the opportunity of having advertisements on the application/website?

Appendix D: The use Case Diagrams

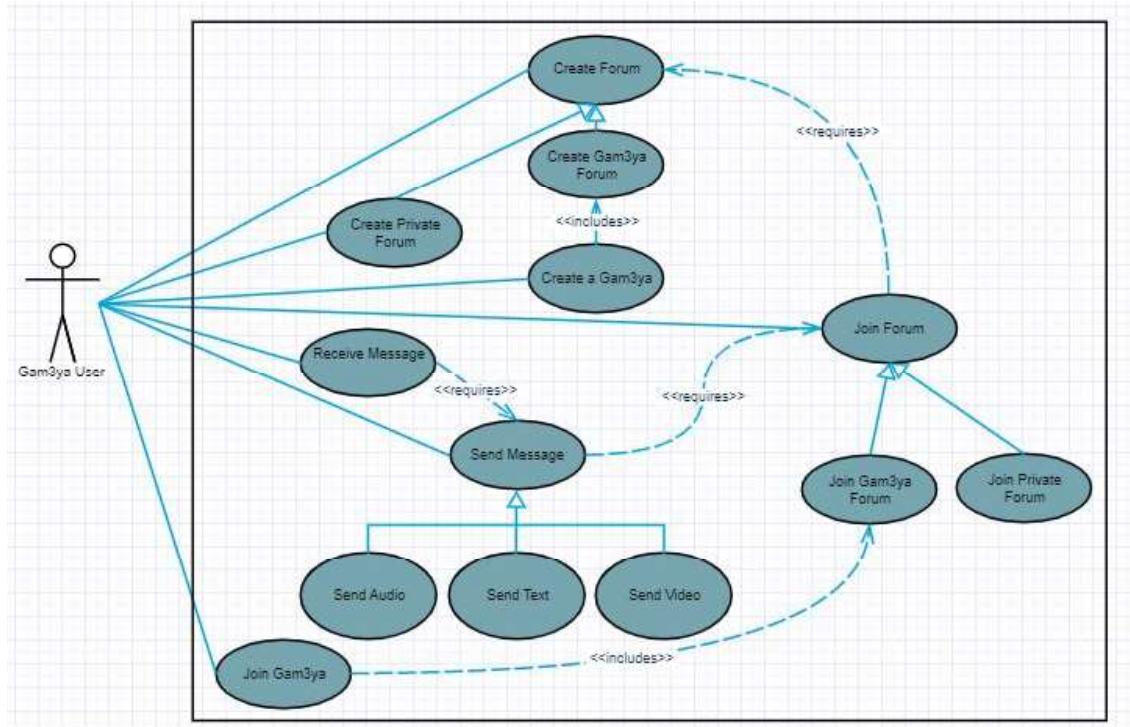
Create an Account



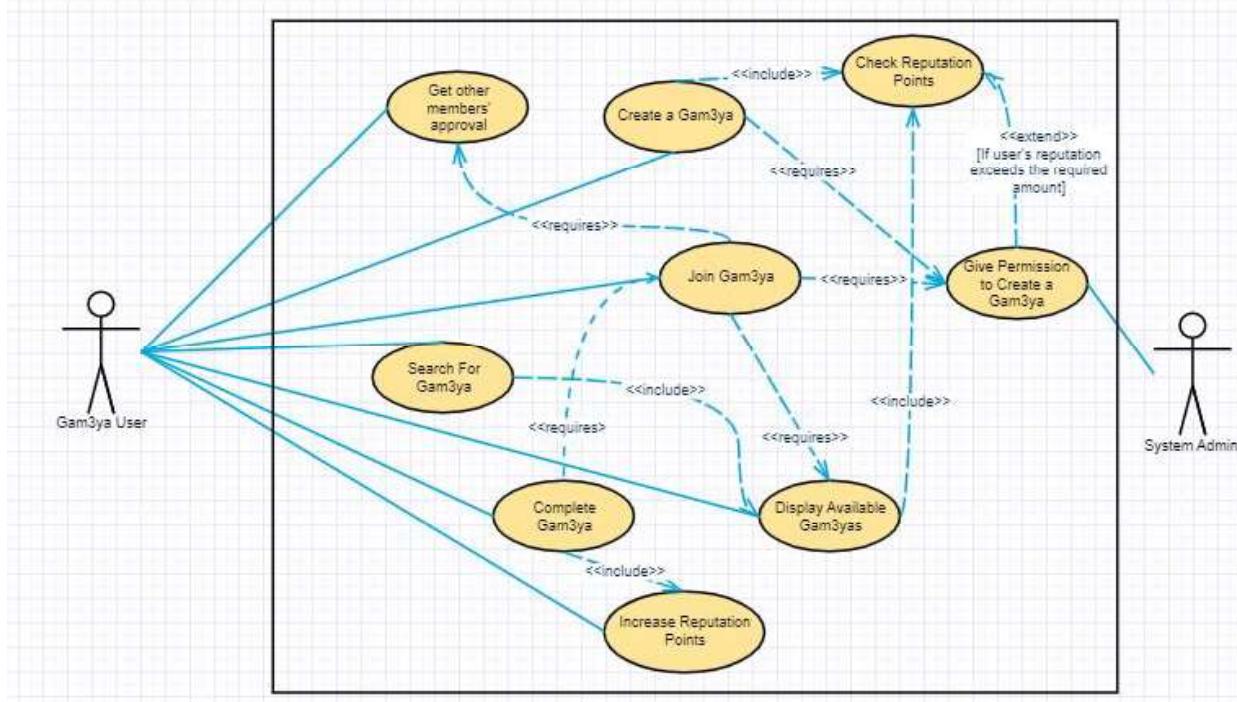
Payment subsystem



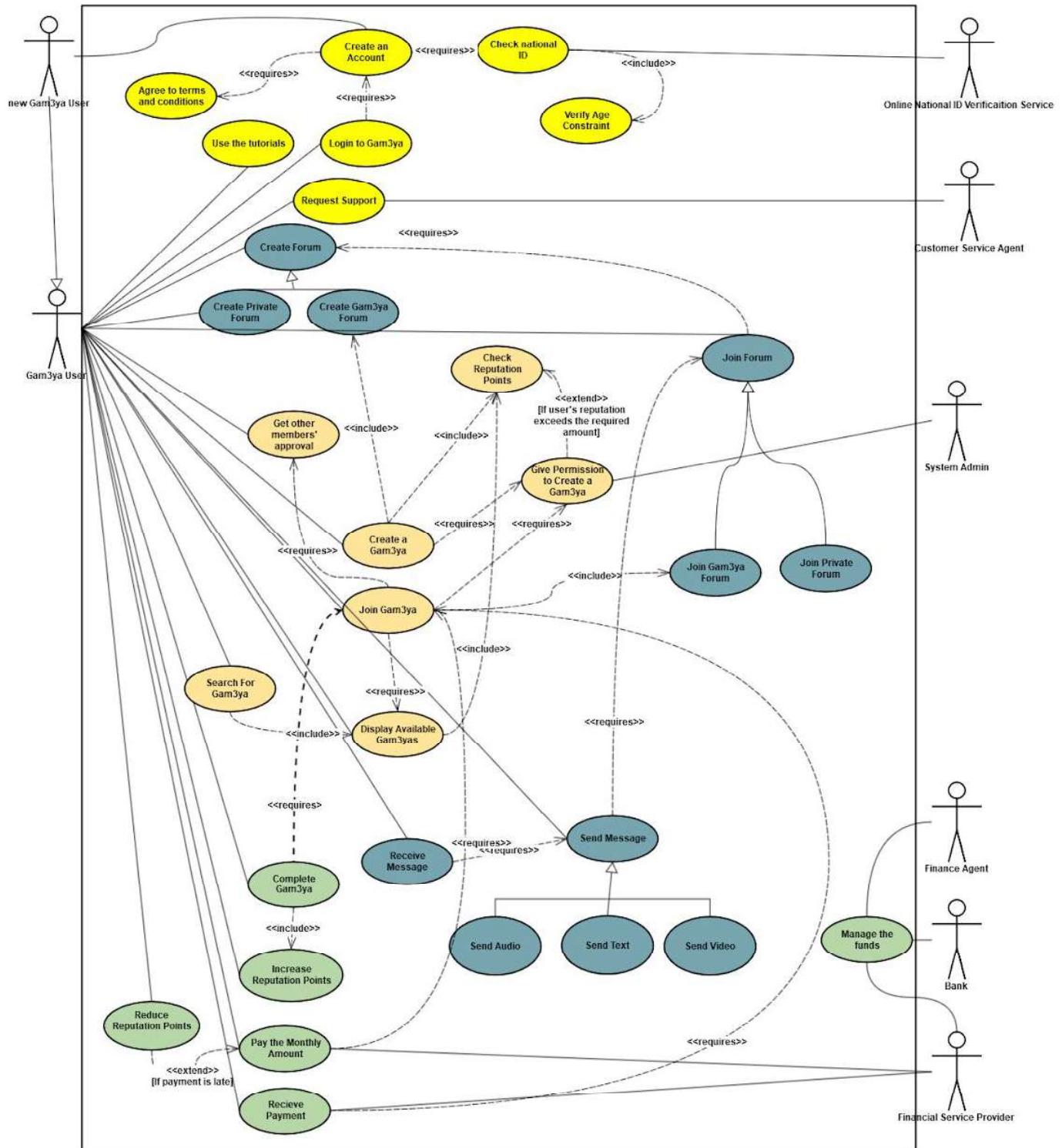
Chatting system



Create/Join a Gam3ya

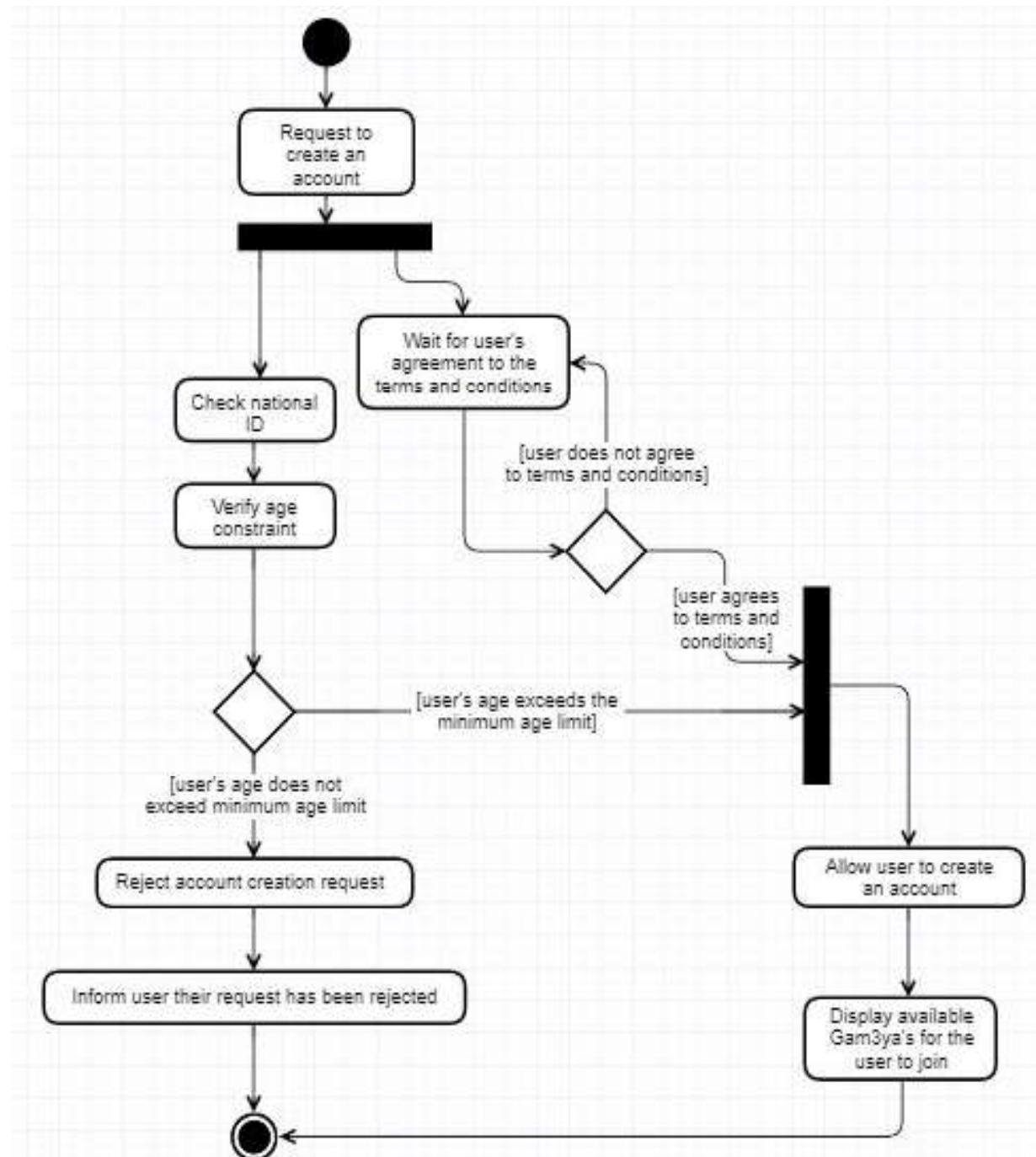


The full Use Case Diagram

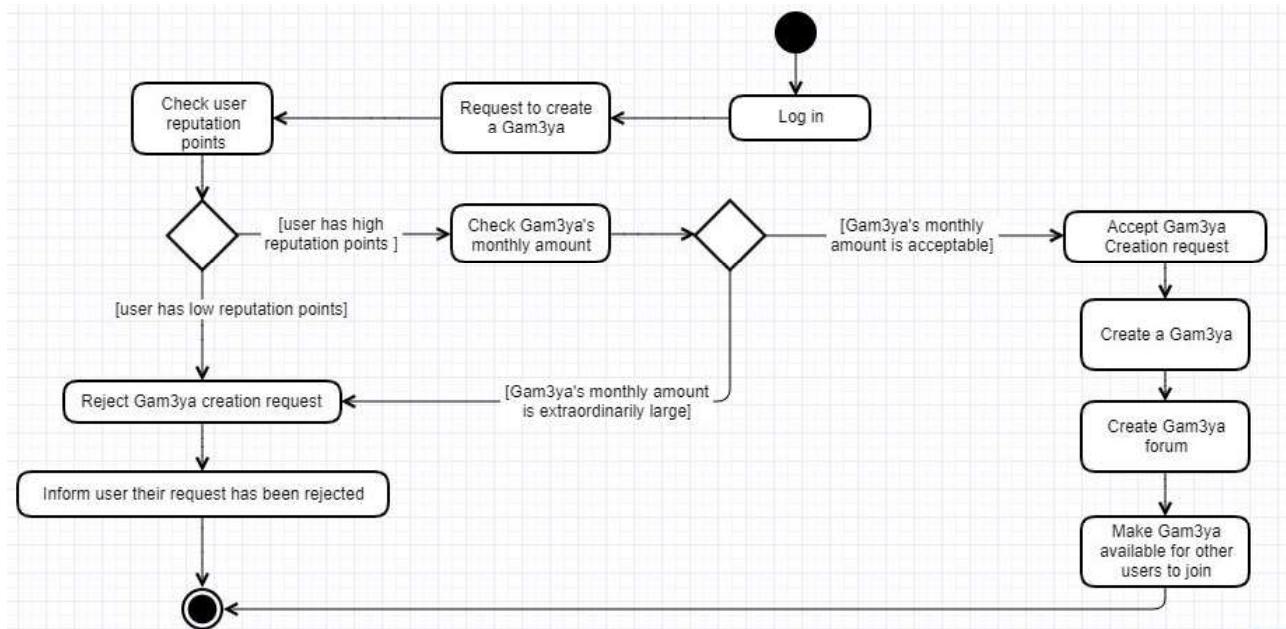


Appendix E: The Activity Diagrams

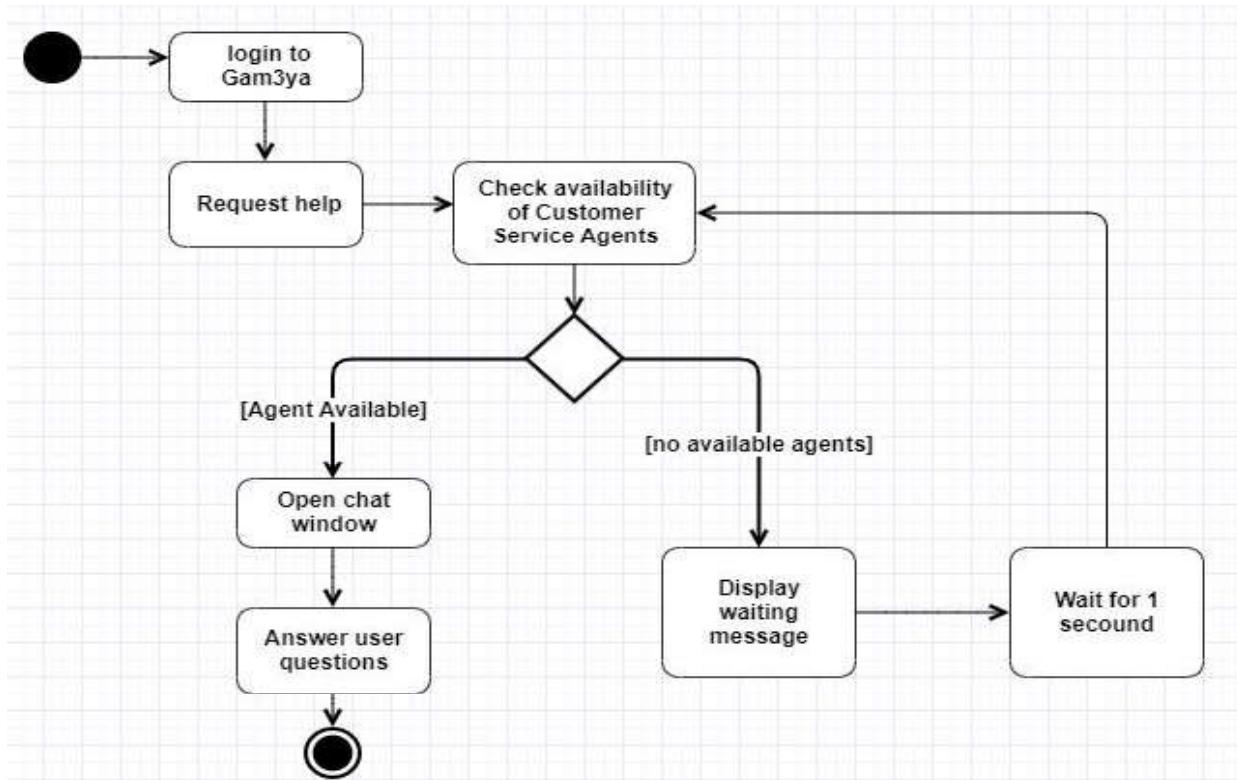
Create an account



Create a Gam3ya



Request Help



Join a Gam3ya

