

Analysis/Design Document

Matthew Hutchinson – Project Manager

Chris MacEachern – Analyst/Developer

Nimna Ekanayaka – Analyst/Developer

CIS-2261

January 18th, 2016

Client Sign Off

Table of Contents

Contents

[Client Sign Off 2](#_Toc441744297)

[Table of Contents 3](#_Toc441744298)

[Management Summary 5](#_Toc441744299)

[System Goal Statement 6](#_Toc441744300)

[Deliverables 7](#_Toc441744301)

[Assumptions and Constraints 9](#_Toc441744302)

[Key System Requirements 10](#_Toc441744303)

[System Modeling 11](#_Toc441744304)

[Use Case 1: Login 14](#_Toc441744305)

[Use Case 2: Registration 18](#_Toc441744306)

[Use Case 3: Join chat 22](#_Toc441744307)

[Use Case 4: Join group chat 26](#_Toc441744308)

[Use Case 5: Send private message 30](#_Toc441744309)

[Use Case 6: sends message 34](#_Toc441744310)

[Program Design 38](#_Toc441744311)

[Register 40](#_Toc441744312)

[Request chat 41](#_Toc441744313)

[Request Group Chat 42](#_Toc441744314)

[Private chat 43](#_Toc441744315)

[Send message 44](#_Toc441744316)

[Package diagram 45](#_Toc441744317)

[Output Design 46](#_Toc441744318)

[Input Design 47](#_Toc441744319)

[Database Design 48](#_Toc441744320)

[User 49](#_Toc441744321)

[Message 50](#_Toc441744322)

[Group Chat 50](#_Toc441744323)

[Support Processing Design 51](#_Toc441744324)

[Environmental Requirements 52](#_Toc441744325)

[Implementation Requirements 54](#_Toc441744326)

Management Summary

Chat Away is designed to allow users to have anonymous conversations with other users in their area. Chat Away is planned to have the following key features:

* The ability to login/register and account.
* The ability to request a 1on1 random chat with a stranger in your area.
* The ability to request/create a group chat for strangers in your area.
* The ability to add random users to a friend’s list and private message them.

This document will explain in detail the key requirements of the system as well as giving a detailed explanation and multiple visual diagrams for the applications primary use cases. This document will also cover the design of the system, showing the program design, database design, as well as the design of system inputs and outputs.

System Goal Statement

The goal of the system that is being developed is a social networking app called Chat Away. Chat Away is a mobile app that will be available for Android devices. The app will allow for a unique way for users to get in contact with one another. Chat Away will allow for users to register for an account and then be entered into a conversation with another user chosen at random. The random conversations will connect users based on area with the users being in the same region. Additionally users will also be able to join group chats based on area and also send private messages to specific users.

Another aspect of the system will be with administration. Admins will have the ability to block users and access different chat activity. The ability to generate reports is another feature that administrators will have. Exception reports of user accounts that have not been used over a certain amount of time can be generated. Summary reports will also be generated detailing information on users and different areas.

Along with the development of the actual mobile application the System will also include a database that will store user information and hold message and chat information. The mobile app will have the ability to communicate with the database in order to retrieve both user information and messages.

Deliverables

Chat Away is designed to connect people around the world anonymously to chat with each other. It is an Android application that lets users create chat rooms and let them connect and chat with each other. The messages will be sent to the users as well as will be saved in the database for later access.

**Analysis Deliverables:**

* Group Meetings: Meet with group members and brainstorm new ideas.
* Group Interviews: After meetings, conduct a series of interviews select absolute requirements.
* Analyze Software Resources: Review Software requirements such as Java, MySQL and PHP and install and maintain them.
* Design weekly status report: Design weekly status reports to hand out to the members.
* Testing documents: Analyze documents for visible errors.
* Illustrate System Design: Create UML diagrams for system activities.
* Review Company Records: Verify data consistency.

**Design and Development deliverables:**

* Database design: Create tables based on entity relationship diagrams and normalize them.
* Interface design: Design output and user interfaces.
* Application architecture: Ensure application is scalable, reliable, available, and manageable.
* Network topology: Determine if a physical or wireless network should be used.
* Implement System: Perform a data and system change over.
* Database.
* Working application (go over what the application includes)

**Non-Deliverables:**

* System maintenance: System will need an IT Consultant to maintain the performance.
* System security: Chat Away needs to protect their data integrity.
* Hardware supply: Chat Away as a company is responsible for acquiring the hardware.

Assumptions and Constraints

**Assumptions:**

* User will have mobile device that meets minimum system requirements
* A server that is able to handle the maximum level of traffic will be used
* Proper administration of the app will allow for a safe environment for users to socialize
* App will be developed exclusively for Android

**Constraints:**

* Some areas will have smaller number of users than others
* Less customization will be allowed for non-paying users
* Resources are limited since the development team only consists of 3 people
* Less can be shown on interface since app is being developed for mobile

Key System Requirements

Key user inputs

* Login/registration for users
* Messages
* Blocking users

Key system outputs

* Exception report for inactive users
* User activity summary report

Data managed by system

* Account information
* Messages
* Last login
* Username + password
* Ban timer
* User GPS coordinates

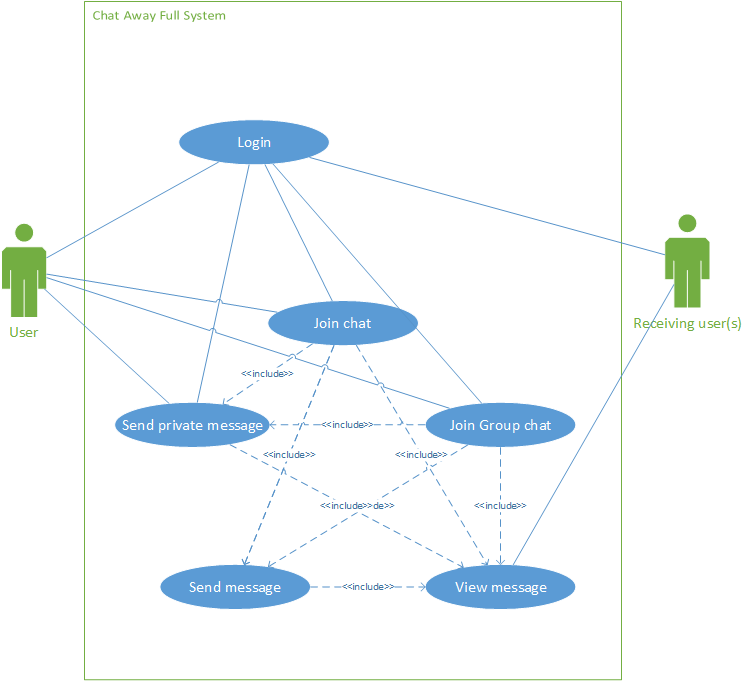
Processes

Key security requirements

* Securely track and transmit account info
* Securely transmit messages

System Modeling

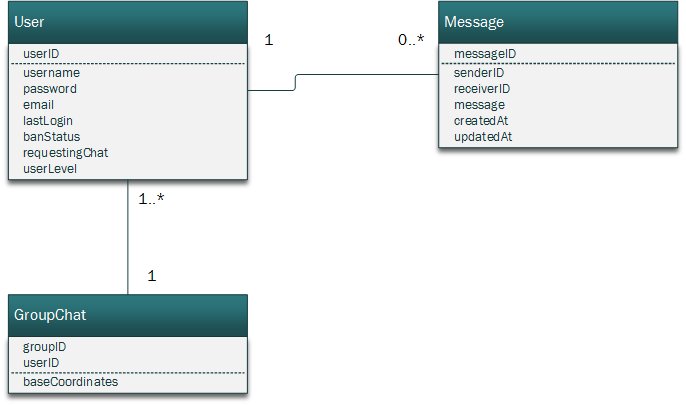
The following diagram depicts the entire system's primary use cases and how they relate to each other. In this diagram “join chat”, “join group chat”, and “send private message” are all dependent on the login use case, showing that a user must be logged in to do any of these things. “Join chat” and “join group chat” use cases include “send private message” as a user is able to send a private message directly from these chat rooms. “View message” is included in all chat related use cases as users can passively view messages during a chat. “Join chat” and “join group chat” also include “send message” which allows the user to send a message to the chat window.



The following is the event table for our system. It contains a list of all system events, the event trigger, source, use case, response, and destination.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Use case** | **Response** | **Destination** |
| User attempts to login | Login attempt | User | Login | Credentials accepted or declined | User |
| User attempts to join chat | join chat attempt | User | Join chat | User is connected to chat if possible | User |
| User registers new account | Account registration | User | Registration | User is registered | User |
| User sends private message | private message is sent | User | User sends private message | message is sent | Selected user |
| User sends message | message is sent | User | User sends message | Message is sent | Connected User(s) |
| User is ban | admin bans user for set amount of time | Admin | ban user | User is ban for X amount of minutes | User |
| User attempts to join group chat | join group chat attempt | User | Join group chat | User is connected to group chat if possible | User |

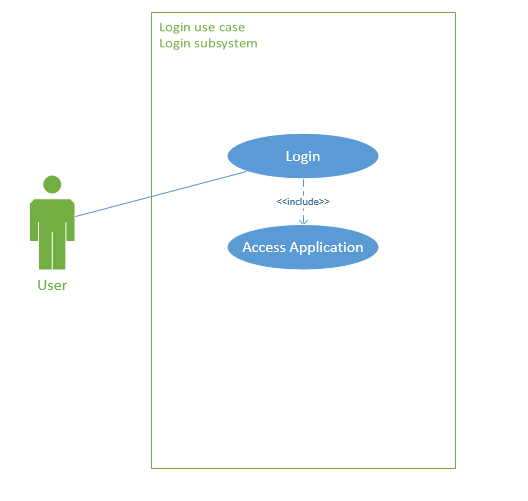
This is the class diagram for our system. In this diagram the User's, Message, and GroupChat classes are shown along with their multiplicity. This diagram shows that each user can send 0 or many messages. Group chat also contains 1 or more users.

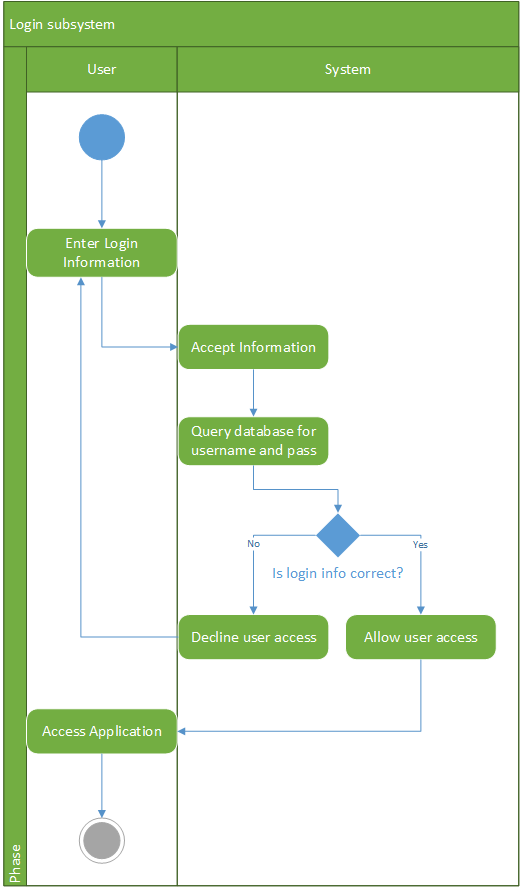


Use Case 1: Login

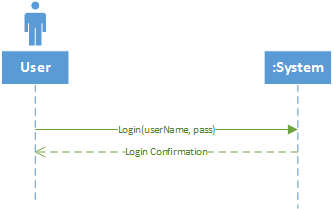
|  |  |  |
| --- | --- | --- |
| Use case name: | Login user | |
| Scenario: | A user attempts to log in | |
| Triggering event: | Login attempt | |
| Brief description: | A user attempts to log into the application and the application verifies their user information | |
| Actors: | User | |
| Related use cases: |  | |
| Stakeholders: | User | |
| Preconditions: | User must exist in the database. User information must be correct | |
| Postconditions: | Member is able to access the application | |
| Flow of activities: | **Actor**  1. User enters login information and attempts login          3.1 User is able to access the system or is re-prompt for username and pass | **System**  1.1 Application queries database for pass based on username    2. Application checks input pass vs. database pass    3. Application either allows entry or denies |
| Exception conditions: | Username isn’t found in database  Password is incorrect | |

This is the use case diagram for the “login” use case. In the diagram the user interacts with the Login use case. Login includes “access application” as login allows access to the application.



This is the activity diagram for the login subsystem. In this diagram the user attempts to login, the system takes the information, checks it against the user database and either allows access to the application or declines access.

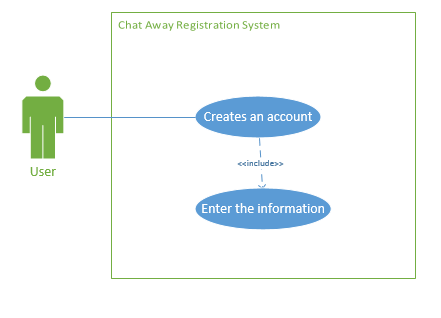
This is the system sequence diagram for the login subsystem. In this diagram the user attempts to login to the system with a username and password and the system either confirms or declines access to the application.



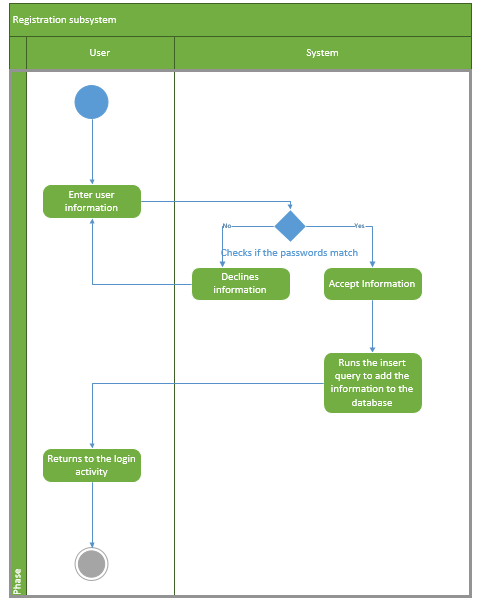
Use Case 2: Registration

|  |  |  |
| --- | --- | --- |
| Use case name: | Register | |
| Scenario: | User registers with the system | |
| Triggering event: | Registration attempt | |
| Brief description: | A user attempts to register with the system and application confirms the registration if there are no conflicting information | |
| Actors: | User | |
| Related use cases: | Login | |
| Stakeholders: | User | |
| Preconditions: |  | |
| Post conditions: | User will be added to the database for login. | |
| Flow of activities: | **Actor**  1. User enters their information. | **System**  1.1 Application checks the two passwords entered by the user and if they match let the information through.  1.2 Application takes the information and writes it to the database. |
| Exception conditions: | Passwords don’t match. | |

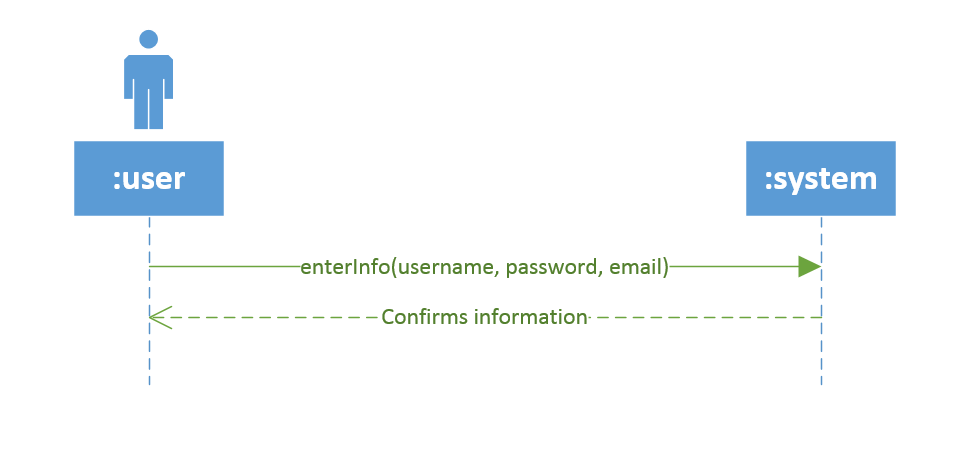
This is the Use Case diagram for registering a user. User creates an account by entering the information.



This is the activity diagram for the register subsystem. User enters information and the program checks if there are conflicting information by checking the two passwords they enter. If the passwords match all the information entered to the form will be sent through a query that enters them to the database and return the user to the login activity.



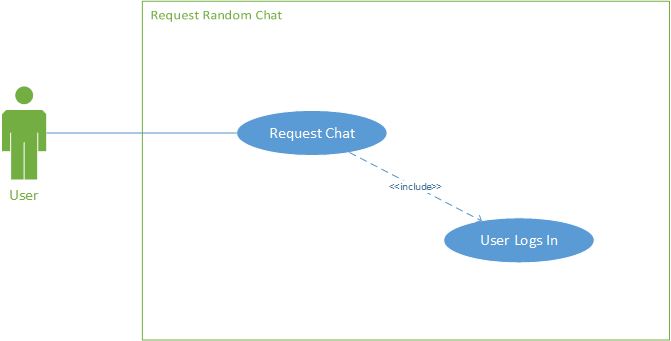
This is the system sequence diagram for user trying to register with the system. User’s information is fed to the system. If there are no conflicting or duplicate information system confirms the information.



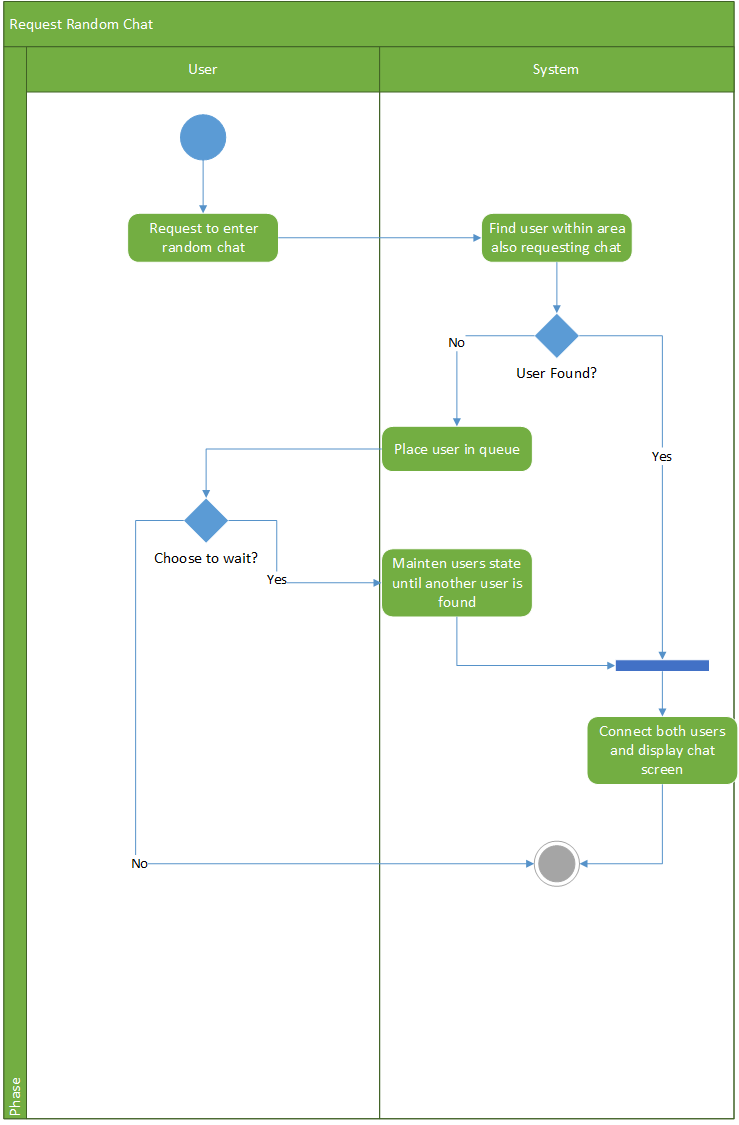
Use Case 3: Join chat

|  |  |  |
| --- | --- | --- |
| **Use case name:** | User requests random chat | |
| **Scenario:** | A user initiates a request for a random chat | |
| **Triggering event:** | User requests a random chat | |
| **Brief description:** | A registered user requests to participate in a random anonymous chat. | |
| **Actors:** | User | |
| **Related use cases:** | User Logs In | |
| **Stakeholders:** | User | |
| **Preconditions:** | User must be registered with an account  Another user must also be available for a chat | |
| **Postconditions:** | Available spots in class is reduced by one  Member account schedule is updated | |
| **Flow of activities:** | **Actor**  1. User selects to be entered into a random one on one chat  2. User is entered into chat | **System**  1.1 System takes the user ID and area and randomly selects another user  1.2 System joins the 2 users |
| **Exception conditions:** | 1.1 No match is found for user or area | |

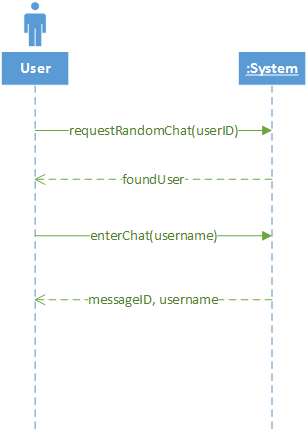
The following diagram is the use case diagram for the user request a random chat user case. The actor in this diagram is the user and the use case “Request Chat” is associated with the user. This use case also includes the “User logs in” use case as a user would need to be logged in to start a chat.



The next diagram is the activity diagram for the request random chat use case. This begins with the user requesting to be entered into a random chat. The system would then need to find a user that is also requesting a chat and within the appropriate area. If a user is not found then the user would be entered into a queue to wait for a chat. The user would be asked if they want to wait. If they choose no then the activity is completed. If they choose yes then the user’s state is maintained until a chat becomes available. If a User is found then the two people are connected and a chat screen is displayed.



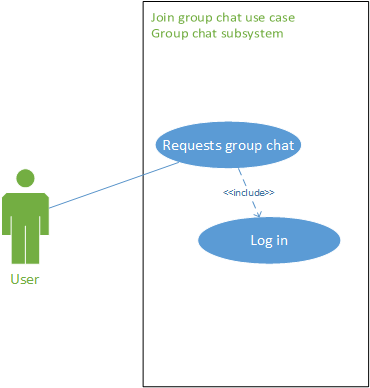
The next diagram is the system sequence diagram for requesting a random chat. The first input to the system is a request for a random chat with the user ID passed to the system. A Boolean of whether a user was found or not is then passed back to the user. The user is then entered into the chat and the user’s username is passed to the system. The message ID and the username is then passed back from the system.



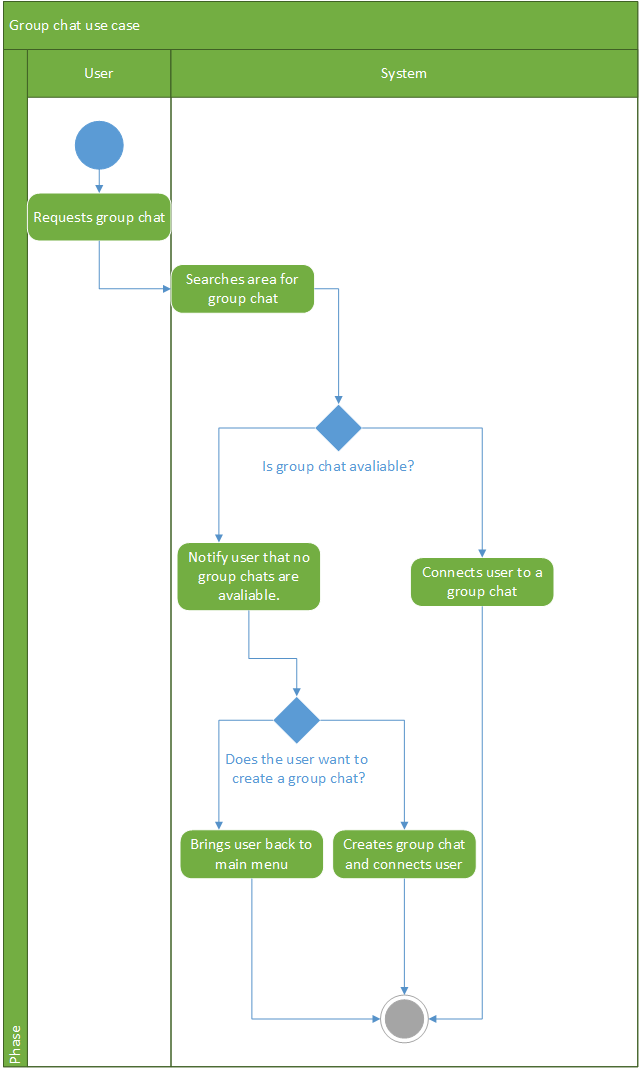
Use Case 4: Join group chat

|  |  |  |
| --- | --- | --- |
| Use case name: | Join group chat | |
| Scenario: | A user wants to join a group chat | |
| Triggering event: | Enter group chat | |
| Brief description: | The user is connected to a group chat and receives current messages | |
| Actors: | User | |
| Related use cases: | Join chat | |
| Stakeholders: | User | |
| Preconditions: | User must be logged in.  User must be connected to the internet.  There must be a group chat available in the users location | |
| Postconditions: | Member is connected to a group chat | |
| Flow of activities: | **Actor**  1. User selects “join group chat” option | **System**  1.1 System checks for closest group chat.  1.2 System connects user to group chat |
| Exception conditions: | No group chats are found. | |

This is the use case diagram for the “join group chat” use case. In this diagram the user interacts with the “request group chat” use case. The “request group chat” use case includes the “log in” action as a use must be logged in to join a group chat.



This is the activity diagram for the “join group chat” use case. In this diagram a user requests to join a group chat. The system then searched the users area for a local group chat. If a group chat is found the user is connected to the chat. If there are no available group chats in the area the user will be prompted to create one. If the user declines the opportunity they will be taken back to the main menu. If the user chooses to create a group chat the chat will be created and they will be connected.



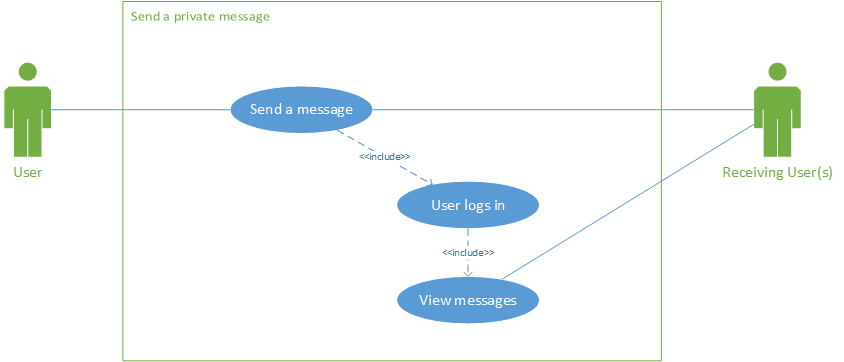
This is the system sequence diagram for the “join group chat” use case. This diagram is very similar to the activity diagram. If a user’s is unable to find a group chat they will be given the option to create a chat.



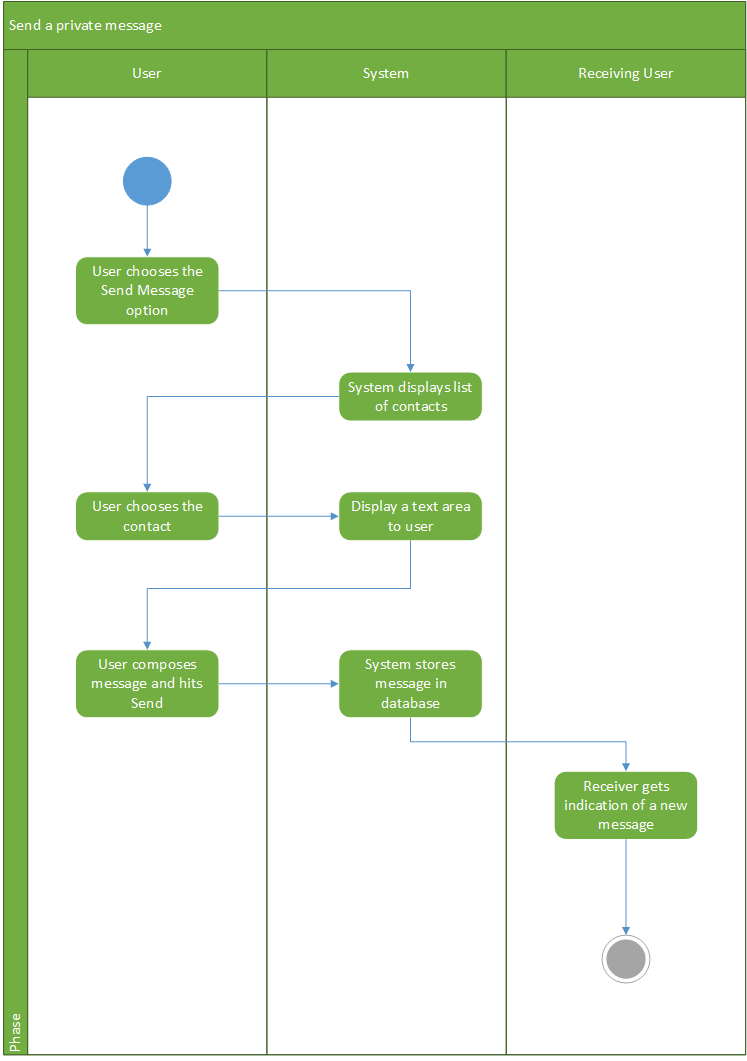
Use Case 5: Send private message

|  |  |  |
| --- | --- | --- |
| **Use case name:** | User Sends Private Message | |
| **Scenario:** | A user sends another user a private message | |
| **Triggering event:** | User chooses to send a private message | |
| **Brief description:** | A user will have the option to send a private message to another user | |
| **Actors:** | User | |
| **Related use cases:** | User logs in | |
| **Stakeholders:** | Users, Admins | |
| **Preconditions:** | User must be registered  The user must not be blocked | |
| **Postconditions:** | Message is stored in database  Receiver gets indication that they have a message | |
| **Flow of activities:** | **Actor**  1. User chooses to send a message    2. User selects the person they want to private message    3. User hits the Send button | **System**  1.1 System displays a list of available users    2.1 System displays user with a text area to compose message    3.1 System adds message to database and the receivers inbox is updated to display an unread message |
| **Exception conditions:** | 1.1 User is not logged in  2.1 The user has been blocked  3.1 Database is unavailable | |

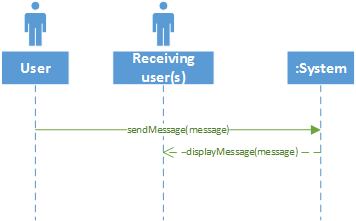
The following diagram is the use case diagram for the “User sends a message” use case. The diagram has one actor who is the user of the Chat Away app. The use case that the user directly interacts with is the “Send a message” use case. This use case also includes “User logs in”.



The next diagram is the Activity Diagram for Send a message. The diagram start with the User who hits the Message button. The system then displays a list of contacts that the user can send a message to. The user then chooses the contact that they want to send the message to. The system displays a text area that the user can compose the message in. Once the user composes the message they hit Send. The system stores the message in the database and the contact is notified of a new message.



This is the system sequence diagram for sending a private message. The sending user sends the message and the system then displays the message to the receiving users.

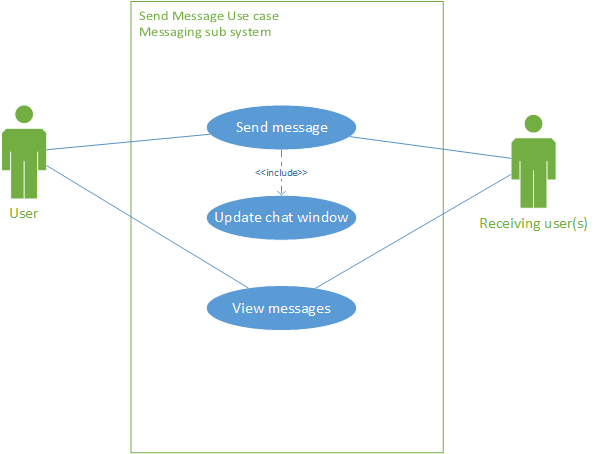


Use Case 6: sends message

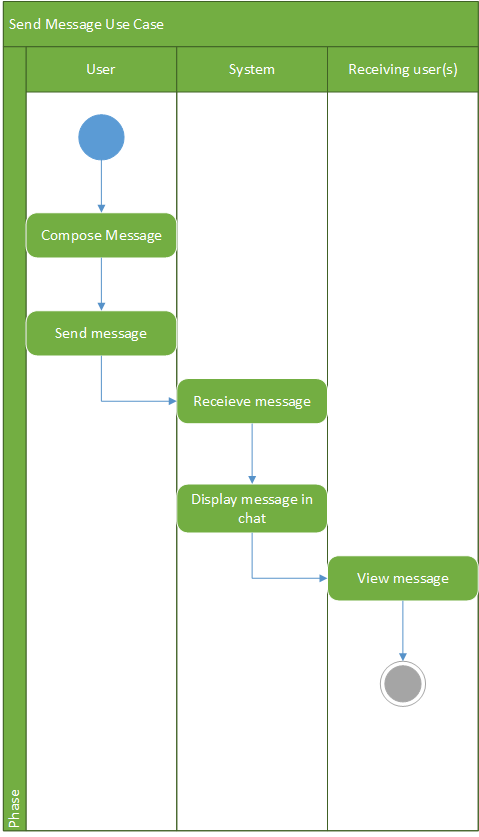
|  |  |  |
| --- | --- | --- |
| **Use case name:** | User Sends Message | |
| **Scenario:** | A user sends a message | |
| **Triggering event:** | User sends a message | |
| **Brief description:** | The user sends a message in either a single or group chat | |
| **Actors:** | User | |
| **Related use cases:** | User logs in | |
| **Stakeholders:** | Users, Admins | |
| **Preconditions:** | User must be registered  The user must not be blocked | |
| **Postconditions:** | Receiver(s) gets indication that they have a message | |
| **Flow of activities:** | **Actor**  1. 1. User types a message and hits send.    2.    3. | **System**  1.1. The system accepts the message and displays it in the chat window |
| **Exception conditions:** | 1.1 User is not logged in  2.1 The user has been blocked  3.1 The user does not have internet access | |

This diagram is the use case diagram for the send message use case. In this diagram there are two actors, a “user” actor and a “receiving user(s)” actor. Because this use case used in both single and group chats it must accommodate multiple receiving users.

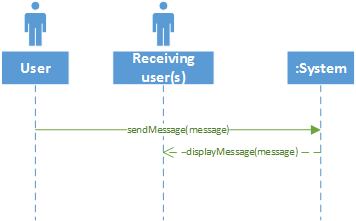
This diagram shows each user having the ability to send and view messages. The “send message” use case includes the “update chat windows” case which is needed for the instant messenger to flow correctly.



This is the activity diagram for the send message use case. This diagram shows the user composing a message, sending it, and the system receiving the message. The system then displays the message in the chat and allows the receiving user to view it.

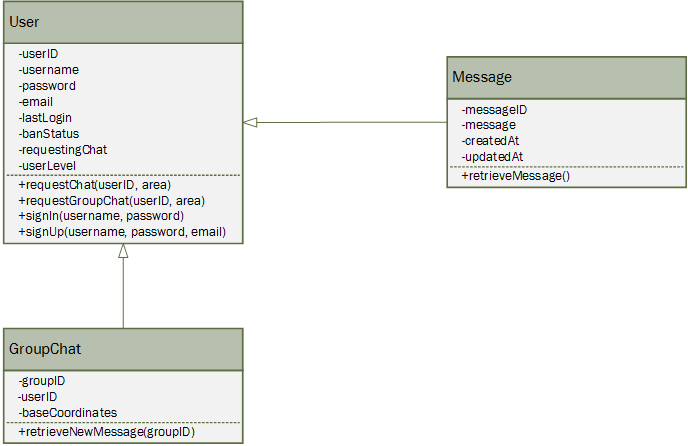


This is the system sequence diagram for the send message use case. Much like the activity diagram you can see the user sending a message to the system and the system displaying it to the receiving user.



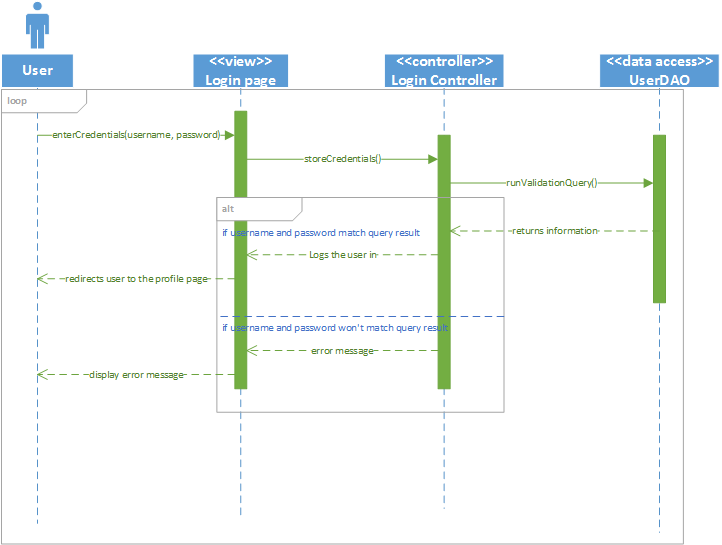
Program Design

Chat Away has been carefully designed for simplicity. In this diagram you can see the class diagram for the Chat Away application. This diagram shows the user, message, and group chat classes and their attributes and methods.



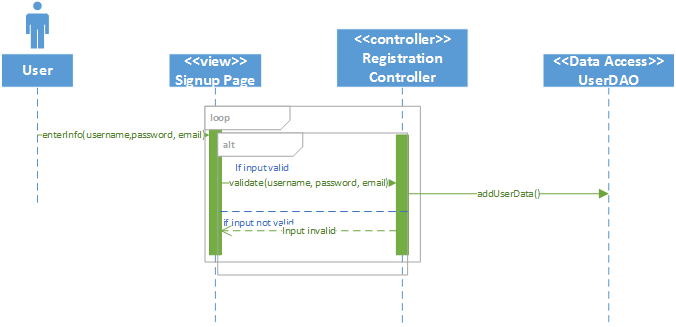
**Login**

This is the full sequence diagram of a user logging in to the app. User enters their credentials and are stored as an object in the business layer. Later it is passed on to the data layer where query is ran with the object data. Query returns the username if the object data are valid. If it is valid user is logged in to the system with a greeting message.



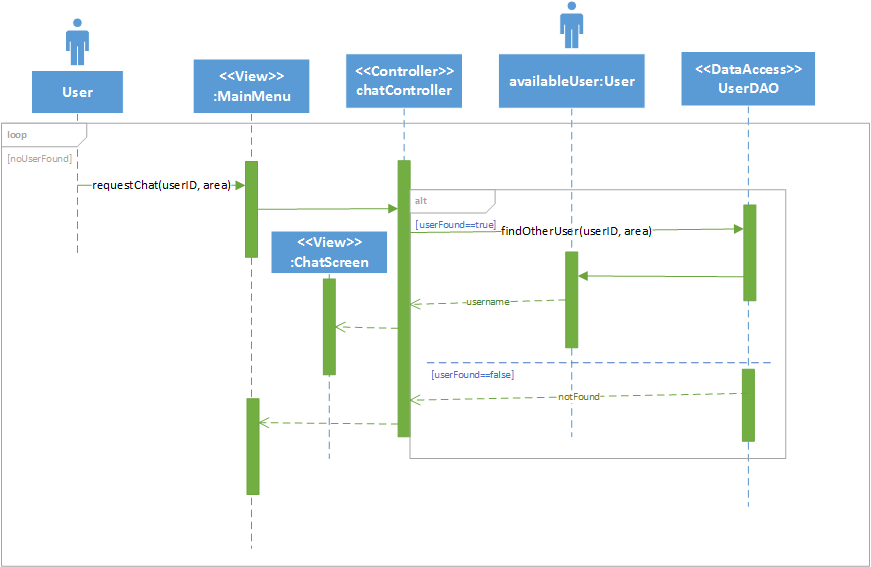
Register

Below is the full system sequence diagram of user registering with the Chat Away. User enters username, password and email. There are two password fields validating user’s input. If the passwords match user object is thrown into the insert query which will write it to the database.



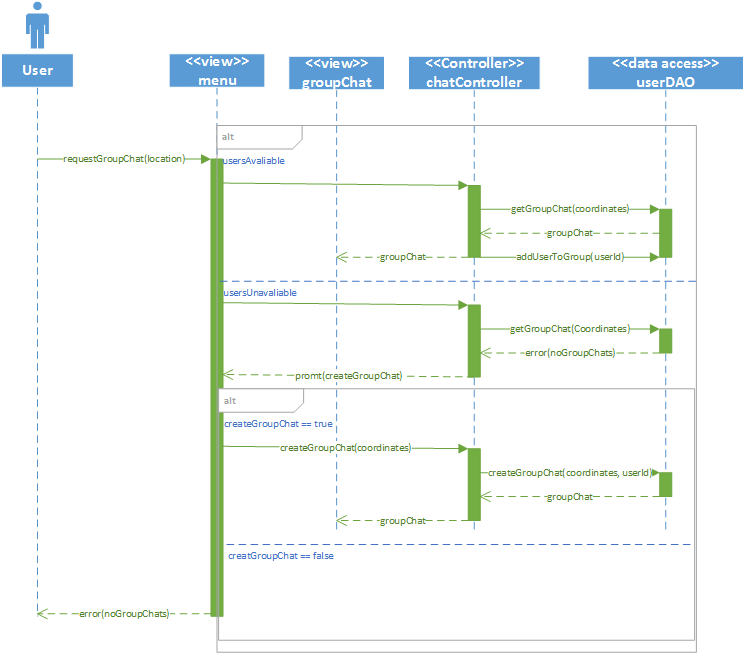
Request chat

The following diagram is for when a user requests a random chat. The request chat method passes the user ID and the area that the user is in. This is done when the user selects the option to join a random chat from the main menu. The chat controller allows for communication to the UserDAO class which is used to search the database for another user seeking a random chat that is within the area. An Alt frame is used to display the two possible events of a user being found or not. If a user is found then the two users are connected in a chat screen. If not then the user is returned to the main menu.



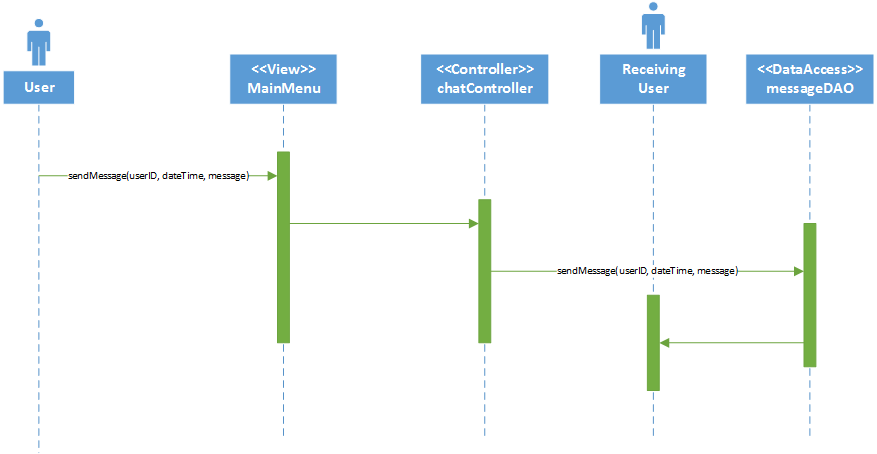
Request Group Chat

The following diagram is the full system sequence diagram for the request group chat use case. In this diagram the user requests a group chat giving their location. The data is then transferred to the business layer and finally the data access layer where if there are any group chats available the user is send to one. If there are no group chats in the user's location the system prompts the user to create a new group chat. If the user decides to create a chat a group chat is created at their coordinates.



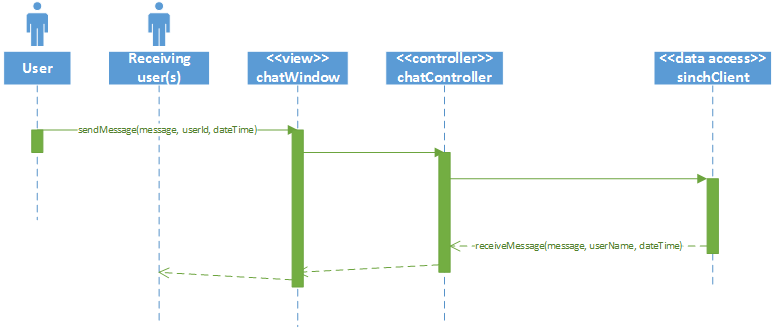
Private chat

This diagram displays the process of a user sending a private message to another user. The “sendMessage” method passes the user ID, the date and the message after the user chooses to send a private message from the main menu. The chat controller calls the messageDAO class in order to store the message in the database. The message is then accessible to the receiving user.

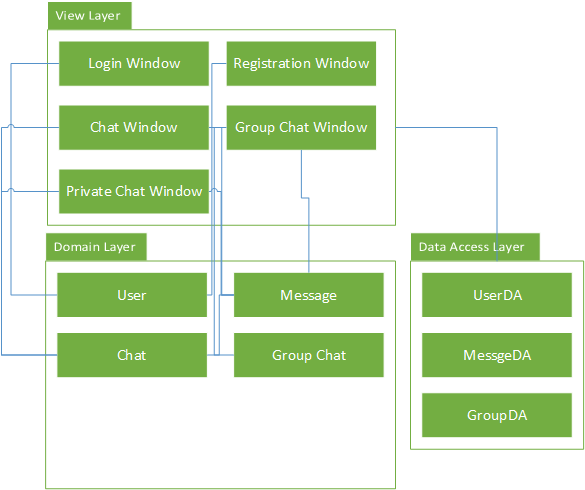


Send message

This is the system sequence diagram for the send message use case. In this diagram the user sends a message, giving their message, userId, and the date/time the message was sent. The data is then passed to the controller and then to the data access layer. The message is then sent to the receiving users.



Package diagram



Output Design

**System Documentation**

NAME OF SYSTEM DATE PAGE 1 OF 1

Chat Away January 27, 2016

ANALYST PURPOSE OF DOCUMENTATION

Chris MacEachern Display information on a specific user

|  |  |  |
| --- | --- | --- |
| FIELD | FIELD TYPE | FIELD LENGTH |
| Username | Alphanumeric | 30 |
| User ID | Numeric | 10 |
| Sign Up Date | Date | 10 |
| Last Login Date | Date | 10 |
| Paid User | Alphanumeric | 3 |
| Ban | Alphanumeric | 3 |

COMMENTS

1. At the top of the report the username, user ID, sign up date and the paid user status is displayed for the specific user the report was chosen for
2. A table shows the total number of messages sent, the last login date, and whether the user is banned.

SORT SEQUENCE

No sort sequence is needed since the information is specific to one user

TOTALS REQURIED

1. The total number of message that the user has sent is displayed

MEDIA

The report is printed on single-ply, standard white stock paper

FREQUENCY

The report is printed at the administrators request when they need information on a specific user

DISTRIBUTION

The report is provided to the administrator at request and further distribution would be decided by the administrator

ATTACHMENTS

Mock-up report is attached

This report is intended to display information about a specific user. This report would be generated if an administrator wished to see information about a specific user. The total number of messages, the last login date, and the ban status are provided.

**Chat Away**

**User Summary**

**Username:** TestUser **User ID:** 9094

**Signup Date:** 01/22/2016 **Paid User:** No

|  |  |  |
| --- | --- | --- |
| **Total Messages** | **Last Login Date** | **Ban Status** |
| 202 | 01/25/2016 | Not currently banned |

**System Documentation**

NAME OF SYSTEM DATE PAGE 1 OF 1

Chat Away January 27, 2016

ANALYST PURPOSE OF DOCUMENTATION

Chris MacEachern Show inactive users

|  |  |  |
| --- | --- | --- |
| FIELD | FIELD TYPE | FIELD LENGTH |
| Username | Alphanumeric | 30 |
| Paid User | Alphanumeric | 3 |
| Sign Up Date | Date | 10 |
| Last Login Date | Date | 10 |
|  |  |  |

COMMENTS

1. The date is included at the top of the report
2. A table includes the list of inactive users. The Username, whether they are a paid user, the sign up date, and last login date are displayed
3. The total number of inactive users is displayed at the bottom along with the date of the last report

SORT SEQUENCE

The claims are ordered in alphabetical order based on the username

TOTALS REQURIED

1. The total number of inactive users is required

MEDIA

The report is printed on single-ply, standard white stock paper

FREQUENCY

The report is printed monthly near the end of every month

DISTRIBUTION

The report should be delivered to the database administrator the day that it is printed

ATTACHMENTS

Mock-up report is attached

This report is an exception report and will display a list of users that have not logged into their account within a certain period of time. The username, paid user status, sign up date and last login date are displayed. A total of inactive users is displayed as well as the date of the last inactive report that was generated.

**Chat Away**

**Inactive Users Report Date:** January 27, 2016

|  |  |  |  |
| --- | --- | --- | --- |
| **Username** | **Paid User** | **Sign Up Date** | **Last Login Date** |
| TestUser | No | 05/27/2015 | 07/3/2015 |
| AnotherUser | No | 02/9/2014 | 06/28/2015 |
| User3 | Yes | 06/22/2014 | 07/25/2015 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

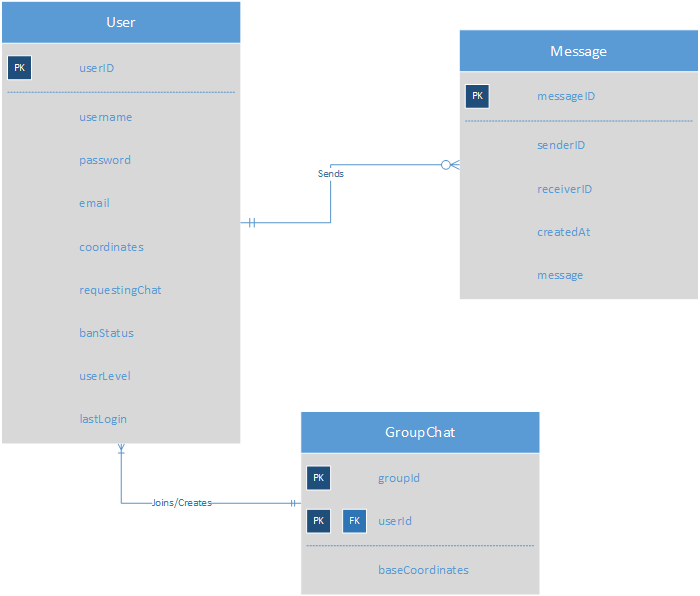
**Number of Inactive Users:** 7

**Date of Last Report:** December 27, 2015

Input Design

Database Design

This is the final ERD of Chat Away. This diagram shows the 3 base classes: user, message, and group chat. Along with their primary keys and foreign keys this diagram also shows the cardinality of the system. Each user can send 0 or many messages. Each user can join/create one group chat at a time, but a group chat has many users.



Found below is the data dictionary for the Chat Away application. This gives an easy to understand view of the database.

User

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | Description | Type | Format | Required | Key | FK Table |
| UserId | User Id | Int (8) | ######## | Y | PK |  |
| Username | Username | Varchar(20) | Xxxxxxxx | Y |  |  |
| Password | Password | Varchar(20) | Xxxxxxxx | Y |  |  |
| Email | User Email | VarChar(30) | Xxxxxxxx | Y |  |  |
| BanStatus | Users current ban status (minutes) | Decimal(8) | ######## | N |  |  |
| RequestingChat | Whether a user is currently requesting a chat | Boolean | True/False | N |  |  |
| AccountLevel | Level of the account | Varchar(10) | Regular/  Advanced/  Admin | Y |  |  |
| Coordinates | Users coordinates based on latitude/  longditude | Varchar(50) | xxxx/xxxx | Y |  |  |

Message

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | Description | Type | Format | Required | Key | FK Table |
| MessageId | Message Id | Int (8) | ######## | Y | PK |  |
| SenderId | User id of the sender | Int (8) | ######## | Y | FK | User |
| ReceiverId | User id of the receiver | Int (8) | ######## | Y | FK | User |
| CreatedAt | Date/Time of creation | Date/Time | Xxxxxxxx | Y |  |  |
| Message | Actual message being saved | Varchar(500) | Xxxxxxxx | Y |  |  |

Group Chat

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | Description | Type | Format | Required | Key | FK Table |
| GroupId | Group Id | Int (8) | ######## | Y | PK |  |
| UserId | User id | Int (8) | ######## | Y | PKFK | User |
| BaseCoordinates | The coordinates the group was created at. | Varchar(50) | xxxx/xxxx | Y |  |  |

Support Processing Design

The only form information submitted to the Chat Away application is the login/registration submission. These forms can be found in the “input design” section of this document.

The ongoing support of the Chat Away system is currently not planned but will be taken into consideration. To continue, long term, a few changes must be made to the application as well as other considerations.

* The database must be moved to a more permanent server where it can support much more information.
* Payments to “sinch” will have to be made, as using their instant messaging client is only free under 1000 users.
* A permanent payment system will have to be added to force users to pay for “advanced” membership.

Environmental Requirements

The following section breaks down the different environmental requirements for the Chat Away system that is being developed. The fact that this is a mobile app that is being developed there are some unique requirements along with some more standard requirements that would be found in any system deployment. The hardware, software, staff and facility requirements are as follows.

**Hardware**

Since Chat Away is being developed as a mobile app any user would be required to have a mobile device in order to use the app. At this time the app is only being developed for Android so any mobile device would need to be capable of running Android in order to use the app. Other than the hardware used to actually use the app a database server will need to be used for the database.

**Software**

As stated in the previous section Chat Away is being developed exclusively for Android. Anybody who wishes to use Chat Away would need to have the Android OS installed on their device. Additionally a database will be used that the app will communicate with. A DBMS will need to be used for the database.

**Staff**

Since this is a brand new application that is being developed the required staff would initially consist of the development team that is in charge of creating the app. Another role that will be need is a database administrator who will be in charge of maintaining the database server and ensuring the integrity of the data.

**Facility**

The database servers will need to be located in an area with an air conditioning system that will keep the hardware from overheating. It would be best to have a specific room to keep the servers in order to ensure the equipment is properly maintained.

Implementation Requirements