**EXPERIMENT - 1**

**AIM:** Write a problem statement for Chat Server System.

**THEORY:** A problem statement is a clear concise description of the issues that needs to be addressed by a problem-solving team. It is used to centre and focus the team at beginning, keep the team on track during the effort, and it is used to validate that the effort delivered an outcome that solves the problem statement.

**Problem Statement:** Chat Server System is an instant messaging application based on LAN and not internet. We have to develop an instant messaging application which will be used to collaborate on group objects and will make data transfer more effective. The software performs all the task of a normal instant messaging application but based on peer to peer networking model. This application will also support blue tooth and hotspot connection and the facility to search in the contacts domain will also be provided

The current internet instant messaging applications suffer vulnerabilities like:

* The limitations of the mobile devices and wireless communications.
* Constant need of an internet connection.
* Possibility of hacking of an account as the account is linked with one computer.
* Possibility of getting vulnerabilities through internet such as viruses, malwares and ransom wares.
* There are also some security risks like the content of the instant message may be intercepted. As a result, the sensitive data like customer list, sales report may be revealed on the internet.

To overcome the vulnerabilities of the above internet messaging application we propose an application that is able to perform the following task:

* LAN based messenger software that works on LAN without the need of an internet connection or a dedicated server.
* Storage of chat data and media.
* Efficient communication between users by instant message receipts.
* Prevents unauthorized user access by asking for username and password for login.
* Efficiency in modification, sorting and retrieval of data.
* The system supports the feature of multiple languages.
* Users should be able to transfer of media with proper encryption keeping data safe.
* Chat and media backups should be maintained depending on user convenience in the local storage for better recovery of lost data.
* User should be able to share stickers and emoticons along with text messages.
* The application supports audio and video messages.
* The application supports a unique ID. If the two users accept each other’s request, they can initiate a conversation using their respective ID’s only.
* The user must be able to block or unblock any of the chats based on the requirements.
* The application supports proper encryption based on which a chat can be saved and provided with a cipher and only after the correct key is entered the chat can be viewed at the other end.
* Each person gets to set a profile picture which he/she can set as public or private.
* Each person can add a status which can be set as public or private.
* Each person can start group chat or group conference having more than one participants.

**RESULT:** The problem statement of the system assigned is properly understood and laid out in front of the students. Also, the functionality of the system is described, which will make the further steps in project easier.

**DISCUSSION:** The definition of the problem statement of the system has made a clear picture of the system under consideration, which will help in smooth passage into the next segments of the project.

**CONCLUSION:** The problem statement is produced which contains all the consolidated requirements of the software to be produced.

**EXPERIMENT - 2**

**AIM:** Write the initial requirement document of Chat Server System.

**THEORY:** Information Requirement Document (IRD) is the most basic paperwork produced during the software development. It is written during the inception of the software development process.

The IRD is used to consolidate and compile all the initial requirements of the user/customer. The IRD also consists of other important information about the software, such as Title of Project, Stakeholders Involved, Techniques used to capture requirements, Development Team Members, and the Version of the Software being produced.

Development of IRD has many advantages, such as it specifies all the initial requirements for the software and it also acts as an input to create Use-Case Diagram, which is also an important document produced during software development.

**STEPS TO BE FOLLOWED:**

* The specified format of writing the IRD is looked up.
* The format of the IRD is written on the word file.
* The various sections of IRD are filled accordingly, such as Version, Stakeholders Involved etc.
* The requirements are consolidated and are written under the ‘Consolidated Requirements’ section.

**RESULT:**

INITIAL REQUIREMENT DOCUMENT

|  |  |
| --- | --- |
| **Title** | Chat server system |
| **STAKEHOLDERS** **INVOLVED** | Users, Developers |
| **TECHNIQUES** **USED** **IN** **CAPTURING** **REQUIREMENTS** | Interviews, Brainstorming, Use Cases |
| **DEVELOPERS** | Vaibhav Agrawal  Shrey Gupta |
| **DATE** | 20/09/19 |
| **VERSION** | 1.0.0 |

|  |
| --- |
| **CONSOLIDATED INITIAL REQUIREMENTS-**  1) A new user should be able to register into the system by giving valid name and phone number.  2) Registered User should be able to log in with phone number and password.  3) The System should be able to manage the case when the user enters invalid credentials and an option to change the password via OTP sent to the registered Email id.  4) A registered user must be able to start a conversation with other registered users in his contact list.  5) The system shall support the feature of read receipts to enable the users to communicate effectively.  6) A user should be allowed to back up chats and media in local storage for retrieval at some other time.  7) The user should be allowed to choose the language he/she wants to communicate in.  8) The user should be able to send Emoji and stickers along with text messages to enhance the user experience and make the chat more expressive.  9) The user should be able to send images and video messages via end to end encryption.  10) The user should be able to set a profile picture and change it whenever he/she desires.  11) The user should be able to add status to their Chat profile that can be read by his/her contacts.  12) The user should be able to search in contact domain to find the person he/she wants to interact with.  13) The system shall also support Bluetooth and hotspot connection to the users.  14) The user should be able to initiate group chats or group conferences for interacting with many users at one go.  15) The administrator should be able to produce user reports and the user should be able to read chat records. |

**DISCUSSION:** The Initial Requirement Document of the system has made a picture of the system under consideration, which will help in smooth passage into the next segments of the project.

**CONCLUSION:** The Initial Requirement Document is produced which contains all the consolidated requirements of the software to be produced.

**EXPERIMENT - 3**

**AIM:** To make USECASE Diagram for Chat Server System.

**DESCRIPTION:** The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analysed to gather its functionalities use cases are prepared and actors are identified. Now when the initial task is complete use case diagrams are modelled to present the outside view.

So, in brief, the purposes of use case diagrams can be as follows:

* Used to gather requirements of a system.
* Used to get an outside view of a system.
* Identify external and internal factors influencing the system.
* Show the interacting among the requirements are actors.

Use case diagrams are considered for high level requirement analysis of a system. So, when the requirements of a system are analysed the functionalities are captured in use cases.

So, we can say that use cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

The actors can be human user, some internal applications or may be some external applications. So, in a brief when we are planning to draw a use case diagram we should have the following items identified.

* Functionalities to be represented as a use case
* Actors
* Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system.

**STEPS FOLLOWED:** The following provides an outline of processes to draw an efficient use case diagram.

* Identify all the different users of the system.
* Create a user profile for each category of users, including all the roles the users play that are relevant to the system. For each role, identify all the significant goals the users have that the system will support. A statement of the system’s value proposition is useful in identifying significant goals.
* Create a use case for each goal, following the use case template. Maintain the same level of abstraction throughout the use case. Steps in higher level use cases may be treated as goals for lower level (i.e. more detailed), sub-use cases.
* Structure the use cases. Avoid over-structuring, as this can make the use cases harder to follow.
* Review and validate with users.
* Connect the users to the functionality initiated by them in the system.

**RESULTS:**

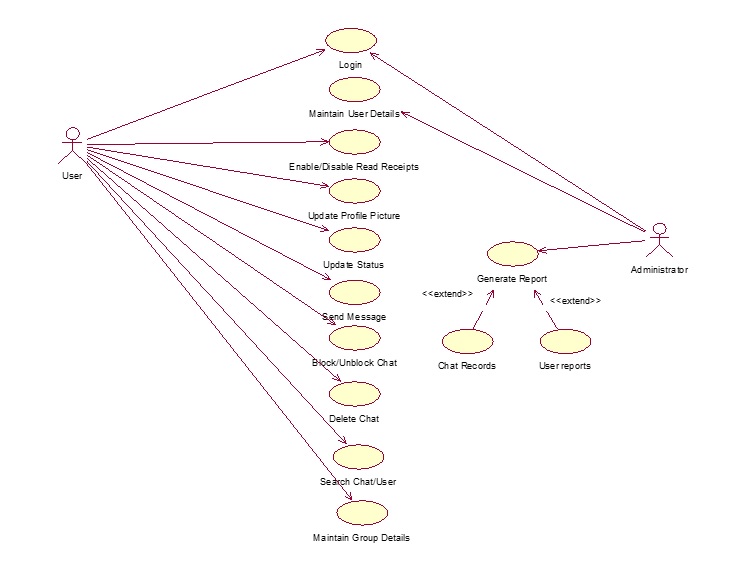
ACTORS:

* User
* Administrator

USE CASES:

* Login
* Maintain user details
* Enable/Disable read receipts
* Update profile picture
* Update status
* Send message
* Block/Unblock chat
* Delete chat
* Search chat/user
* Maintain group details
* Generate report

USE CASE DIAGRAM FOR CHAT SERVER SYSTEM



**DISCUSSIONS:** We learnt to make the USECASE diagram of the “Chat server System” case study by identifying use cases, Actors and relationships between them.

**CONCLUSION:** Use case diagram for chat server system has been implemented.

**EXPERIMENT - 4**

**AIM:** To make USECASE Diagram Description for Chat Server System.

**DESCRIPTION:** Use case description is the detailed explanation of all the use cases defined in the use case diagram. It defines the flow and function of the use case: when will the use case be active, when will it end, what are the necessary conditions to be fulfilled for its functioning.

The following is the template for use case description:

* **Introduction**: Briefly describe the use case.
* **Actors**: List actors that participate and interact with this use case.
* **Pre-Condition:**List pre-conditions here.  List the system state/conditions which must be true before this Use Case can be executed.
* **Post-Conditions:**Post-conditions on a use case lists possible states that the system can be in at the end of the use case execution. The system must be in one of those states. A post-condition also states actions that the system performs at the end of the use case, regardless of what occurred in the use case.
* **Flow of events:**
* Basic flow: primary events on use case execution
* Alternate flow: Any other possible flow.
* **Special Requirements:** Enter any special requirements such as Performance requirements, Security requirements, and user interface requirements.
* **Related use cases:**List the related use cases if any.

The use case description should be in a box.

**STEPS FOLLOWED:**

* Determine all the use cases.
* Take each use case one by one and determine their functionalities and constraints.
* Determine all the requirements for each use case.
* Write the description for each use case in the given standard format.

**RESULT:**

Use Case Description

1. **LOGIN**

|  |
| --- |
| Introduction: This use case allows the user to login to the system. |
| Actors: User, Administrator |
| Pre-condition: The user must have already existing account. |
| Post-condition: The user is successfully logged into the system. |
| Basic flow:   1. The user is required to enter the right credentials i.e. username and password of an already existing account. 2. After the credentials are verified, user is logged into the system. |
| Alternate flow:  Alternate Flow 1: Invalid Details  If the username is invalid, an error message is displayed and user is asked to enter the details again.  Alternate Flow 2: User Exits  User cancels the process and exists. |
| Special Requirements: None |
| Associated use cases: None |

1. **ENABLE/DISABLE READ RECEIPTS**

|  |
| --- |
| Introduction: This use case allows the user to enable/disable the read receipts while chatting. |
| Actors: User |
| Pre-condition: The user must be logged into the system. |
| Post-condition- Read receipts are set according to the user’s preference. |
| Basic flow:   1. The user open settings, selects Account and then selects Privacy option. 2. The user enables/disables read receipts in Privacy settings. |
| Alternate flow:  Alternate Flow 1: User Exits  User exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **UPDATE PROFILE PICTURE**

|  |
| --- |
| Introduction: This use case allows the user to update his/her profile picture. |
| Actors: User |
| Pre-condition: The user must be logged into the system |
| Post-condition: User’s profile picture is successfully updated. |
| Basic Flow:   1. The user taps on current display picture and selects Edit. 2. The user browses the image on his system and selects it as the new profile image. |
| Alternate flow:  Alternate Flow 1: No image on the system  There is no image on the system he is using.  Alternate Flow 2: User exits  The user exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **UPDATE STATUS**

|  |
| --- |
| Introduction: This use case allows the user to update his/her status. |
| Actors: User |
| Pre-condition: The user must be logged into the system. |
| Post-condition: User’s status is successfully updated. |
| Basic flow:   1. The user selects Settings option and then selects Status. 2. The user enters new status and clicks on Save Changes. |
| Alternate flow:  Alternate Flow 1: User exits  The user exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **DELETE CHAT**

|  |
| --- |
| Introduction: This use case allows the user to delete a chat. |
| Actors: User. |
| Pre-condition: The user must be logged into the system. |
| Post-condition: User successfully deletes a chat. |
| Basic flow:   1. The user is allowed to choose a chat to delete. 2. The user opens the settings of the chat and selects the option to delete the chat. |
| Alternate flow:  Alternate Flow 1: User exits  The user exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **BLOCK/UNBLOCK CHAT**

|  |
| --- |
| Introduction: This use case allows the user to block/unblock a chat |
| Actors: User |
| Pre-Condition: The user must be logged on to the system. |
| Post-Condition: Chats are blocked or unblocked according to the user’s preference |
| Basic flow:   1. The user opens the chat to change the settings for. 2. The user opens the settings and chooses to block an unblocked chat or to unblock a blocked chat. |
| Alternate flow:  Alternate Flow 1: User exits  User exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **SEND A MESSAGE**

|  |
| --- |
| Introduction: This use case allows the user to send a message to a group or a personal chat |
| Actors: User |
| Pre-Condition: The user must be logged onto the system. |
| Post-Condition: Information is successfully shared. |
| Basic Flow:   1. The user is allowed to send a message on private chat or on group chat. 2. After choosing the chat user gets the option to send a text message, image, audio message or video message. |
| Alternate flow:  Alternate Flow 1: User exits  User exits the system. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **GENERATE REPORT**

|  |
| --- |
| Introduction: This use allows the administrator to generate reports for chat records and user reports. |
| Actors: Administrator. |
| Pre-Condition: The administrator must be logged onto the system. |
| Post-Condition: Reports are successfully generated. |
| Basic Flow:   1. The administrator can generate report on chat records or user reports.   Basic Flow 1: Chat records   1. The admin retrieves data for a chat. 2. The admin generates a report on number of text messages shared, images shared, video and audio messages shared.   Basic Flow 2: User Report   1. The admin retrieves data for the person. 2. The admin generates report for the user for number of messages shared, number of other users blocked by, number of users blocked, total data used. |
| Alternate flow:  Alternate Flow 1: Admin exits  The admin exits the system. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **SEARCH CHAT/USER**

|  |
| --- |
| Introduction: This use case allows the user to search a chat or a user. |
| Actors: User. |
| Pre-Condition: The user must be logged onto the system. |
| Post-Condition: The search query is successfully executed. |
| Basic Flow:   1. The user chooses the search option to search for a chat or a person. 2. The user types the name of the chat and the search results are retrieved. |
| Alternate flow:  Alternate Flow 1: User exits  The user exits the system. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **MAINTAIN GROUP DETAILS**

|  |
| --- |
| Introduction:This use case documents the steps that the user must follow in order to maintain group details and add, update, delete and view group information. |
| Actors**:** User |
| Pre-Condition:The user must be logged onto the system before this use case begins. |
| Post Condition: If the use case is successful, then the group information is added, updated, deleted or viewed. If not, the system state is unchanged. |
| Flow of Events  Basic Flow   1. This use case starts when user creates a group and wishes to add/delete/update/view group information. 2. The system requests that the user specify the function he/she would like to perform (add user, remove user, update group details, view group details).   Basic Flow 1: Add Participant   1. The system requests the user to enter user information. This includes: 2. Contact No. 3. Name 4. The new participant is then added to the group.   Basic Flow 2: Delete Participant   1. The system requests the user to enter user information. This includes: 2. Contact No. 3. Name 4. The corresponding participant user is deleted from the group.   Basic Flow 3: Update Group Information   1. The system requests the user to enter group information. This includes: 2. Group Name 3. No. of participants 4. Name of all participants 5. Admin Rights 6. The user makes the desired changes to group information. 7. The system updates group information with updated information   Basic Flow 4: View Group Information   1. The system requests the user to enter group information. This includes: 2. Group Name 3. No. of participants 4. Name of all participants 5. Admin Rights 6. The system retrieves the information and displays it to the user. |
| Alternate Flow  Alternate Flow 1: Participant Already Exists  If in Add Participant flow, a participant with same name and contact no. already exists, the system displays an error message. The user returns to basic flow and may re-enter user information.  Alternate Flow 2: User not found  If in Add Participant flow, a participant with specified name and contact no. does not exist, the system displays and error message. The user returns to basic flow and may re-enter user information.  Alternate Flow 3: Update Cancelled  If in Update Group Information flow, user decides not to update the information, update is cancelled and basic flow is restarted at beginning.  Alternate Flow 4: Deletion Cancelled  If in Delete Group Information flow, user decides not to delete the participant, delete is cancelled and basic flow is restarted at beginning.  Alternate Flow 5: User Exits  This allows the user to exit at any time during the use case. The use case ends. |
| Special Requirement: None |
| Associated Use Case: Login |

1. **MAINTAIN USER DETAILS**

|  |
| --- |
| Introduction: This use case enables the administrator to maintain user details. |
| Actors: Administrator. |
| Pre-Condition: The administrator must be logged onto the system. |
| Post-Condition: The user maintains are successfully maintained. |
| Basic Flow:   1. The system requests the administrator specify the function he/she would like to perform (either add a user, update a user, delete a user or view a user). 2. Once the administrator provides the requested information, one of the following sub flows executes.   Basic Flow 1: Add a User   1. The administrator checks for the uniqueness of the user name and the constraints for the password. 2. The user is added to the database and his/her details are maintained and reports are generated.   Basic Flow 2: Update a user   1. The administrator enters the name of the user. 2. The system displays the user information. 3. The administrator makes the required changes.   Basic Flow 3: Delete a user   1. The system requests the administrator to specify the user name. 2. The system displays the information of the user. 3. The system prompts the administrator to confirm the deletion of the user. 4. The administrator verifies the deletion. 5. The system deletes the record.   Basic Flow 4: View a user   1. The system requests the administrator to enter the user name. 2. The system retrieves and displays the user information. |
| Alternate Flow:  Alternate Flow 1: Invalid Entry  If the user name entered does not exist the system displays an appropriate error message. The actor returns to the basic flow and may re-enter the invalid entry.  Alternate Flow 2: User name already exists  If in Add a user flow, a user name already exists then the system displays an error message. The administrator returns to the basic flow and may re-enter the user name.  Alternate Flow 3: User name not found  If in Delete, update or view flow, the user name does not exist, the system displays an error message. The administrator returns to the basic flow and re-enters the user name.  Alternate Flow 4: Update cancelled  If in update a user, the administrator decides not to update a user details, update is cancelled and the administrator returns to the basic flow.  Alternate Flow 5: Delete cancelled  If in update a user, the administrator decides not to delete a user details, delete is cancelled and the administrator returns to the basic flow.  Alternate Flow 6: User exits  This allows the actor to exit the at any time during the use case. The use case ends. |
| Special Requirements: None |
| Associated Use Case: Login |

**DISCUSSIONS:** We learnt to write the USECASE diagram description of the “Chat Server System” case study by identifying use cases, Actors and relationships between them.

**CONCLUSION:** Use case diagram Descriptions for Chat server system has been implemented.

**EXPERIMENT - 5**

**AIM:** To draft the Software Requirements Specification Document for the Chat Server System

**DESCRIPTION:** A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

Draft a Software Requirement Specification Document for the Course Registration System

|  |
| --- |
| 1. Introduction:    1. Purpose    2. Scope    3. Definitions, Acronyms and Abbreviations    4. References    5. Overview |
| 1. Overall Description:    1. Product Perspective   2.1.1 System Interfaces  2.1.2 User Interfaces  2.1.3 Hardware Interfaces  2.1.4 Software Interfaces  2.1.5 Communication Interfaces  2.1.6 Memory Constraints  2.1.7 Operations  2.1.8 Site Adaptation Requirements |
| 1. Specific Requirements:    1. External Interfaces    2. Functions    3. Performance requirements    4. Logical Database requirements    5. Design Constraints       1. Standards compliance    6. Software System Attributes       1. Reliability       2. Availability       3. Security       4. Maintainability       5. Portability    7. Organizing the Specific requirements       1. System mode       2. User class       3. Objects       4. Feature       5. Stimulus       6. Response       7. Functional Hierarchy |
| 1. Supporting Information: |

1. **Introduction**

This document aims at defining the overall software requirements for ‘LAN based Chat Server System’. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing/using this product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

**1.1 Purpose**This specification document describes the capabilities that will be provided by the software application ‘Chat Server System’. It also states the various required constraints by which the system will abide. The intended audience for this document are development team, testing team and end users of the product.

**1.2 Scope**

The name of the software is ‘Chat Server System’. The system will be referred to as CSS in the rest of the SRS. The proposed CSS must be able to perform the following functions.

Do’s:

1. The system is LAN based and does not require internet connection.
2. Issue of login ID and Passwords to system operators.
3. Maintain Details of all the users registered with the system.
4. Start a private or group chat.
5. Send text messages, images, audio and video messages.
6. Update status.
7. Update profile picture.
8. Block/Unblock chats.
9. Enable/Disable Read receipts.
10. Generate reports:
    1. Generate user reports.
    2. Generate chat records.

Don’ts:

1. The system does not support communication over internet.
2. Audio and video calling is not supported.
3. Status does not support images and video stories.

Benefits:

1. The system is well secured and chats can’t be intercepted.
2. The system is free from vulnerabilities of the internet.
3. Efficient and secure sharing of media.
4. Generating of reports.

**1.3 Definitions, Acronyms, and Abbreviations**

1. SRS- Software requirement specification.
2. CSS- Chat server system.
3. DBA- Database administrator.
4. LAN- Local area network.
5. Admin- Data Administrator.

**1.4 References**

IEEE Recommended Practice for Software Requirement Specifications- IEEE Std 830-1998. Object Oriented Software Engineering- Yogesh Singh, Ruchika Malhotra.

**1.5 Overview**

The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail

**2. Overall Description**

The CSS enables the user to send and receive messages, pictures, videos and files without the use of internet. The service is LAN based. It is assumed that the admin has all the rights. The admin maintains the database and generates reports.

**2.1 Product Perspective**

The software is windows based independent software that used Local Area Network to transfer data. The LSS shall be built using client/server architecture and will be compatible with Microsoft Windows Operating System. The front-end of the system will be developed using Visual Basic 6.0and backend will be developed using MY SQL Server 2005.

Front End Client Application (with search/chat user, create new group, start chat, update status, update profile picture, block/unblock chat.

Backend   
Database

**2.1.1 System Interface**

None

**2.1.2 User Interface**

The application will have a user friendly and menu based interface. Following screens will be provided:

1. A Login screen for entering the username, password will be provided. An unregistered user can select his role (User/Admin) and register in the system. Access to different screens will be based on the role of the user.
2. There will be a screen to display all the previous chats of a user. The user can select a chat to send a message.
3. There will be a settings screen where the user can update his/her profile picture and status, choose the language to communicate in, backup chats and enable/disable read receipts.
4. There will be a screen for each chat which displays all the past messages sent and received. This screen also shows the option for displaying all the media content shared. From this screen the user can also block or unblock a chat.
5. There will be a screen for capturing and displaying information about all the users registered on the system.
6. There will be a screen displaying all the information about user chats.

**2.1.3 Hardware Interfaces**

1. The System must run over a local area network (LAN) and all the computers connected to the same network can share data.
2. Screen Resolution of at least 800X600- required for complete viewing of screens. Higher Resolution will not be a problem
3. Support for printer (dot matrix/desk Jet/inkjet) that is, appropriate drivers are installed and computer connected printer will be required to print the reports by the DBA and invoice by customer.
   * 1. **Software Interfaces**
        1. Chat Server System application.
        2. Any windows (Windows XP/VISTA/7/8/10) operating system.
        3. MS Access 2010 as the DBMS –for database will be required on the system used by the DBA. Future Applications will aim at upgrading to Oracle 8i as the DBMS.
        4. Crystal Reports 8 – for generating and viewing reports
     2. **Communication Interfaces**

None

* + 1. **Memory Constraints**

1. At least 64 MB RAM must be available at all times to ensure smooth functioning of the software.
2. At least 100 MB of storage space on external storage.
   * 1. **Operations**

This product release will not cover any automated housekeeping aspects of the database. The DBA at the client site will be responsible for manually deleting old/non-required data. Database backup and recovery will also be handled by the DBA.  
However the system will provide a ‘RESET SYSTEM’ function that will delete (upon confirmation from DBA) all the existing information from the database.

* + 1. **Site Adaptation Requirements**

The terminals at the client site will have to support the hardware and software interfaces specified in above sections.

1. **Specific Requirements:**

This section contains the software requirements in detail along with the various forms to be developed.

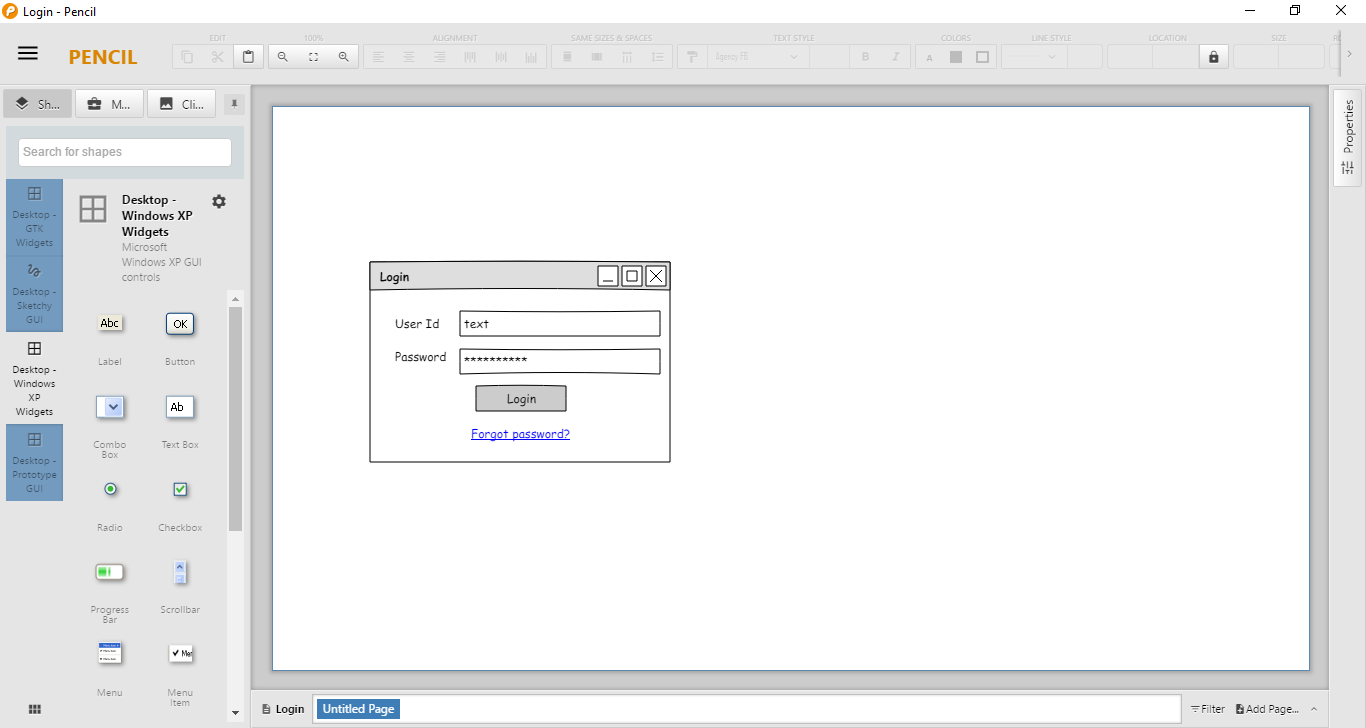
**3.1 External Interfaces**

* + - 1. **User Interfaces**

The following screens will be provided.

1. **Login**

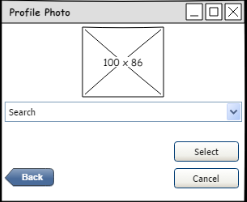
This form will be accessible to system administrator and user. This form will allow registered administrator and user to Login into the system.



Various fields available in this screen will be:

1. User Id: Alphanumeric of length of exactly 10 characters. Special characters other than underscore are not allowed. The system shall not allow the user to enter non-alphanumeric characters and out of range values.
2. Password: Alphanumeric of length up to 8 characters. The system shall not allow the user to enter non-alphanumeric characters and out of range values.
3. **Update Profile Picture**

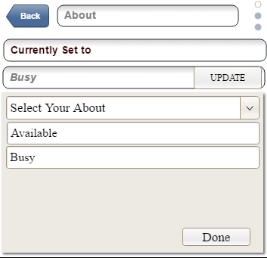
This screen will be accessible to the user. This screen will allow the user to update his/her profile picture.



Various fields in the screen are:

1. Search: Directory path allows the user to browse a picture from the local storage.
2. **Update Status**

This screen will be accessible to the user. This screen will allow the user to update his/her status.



Various fields in the screen are:

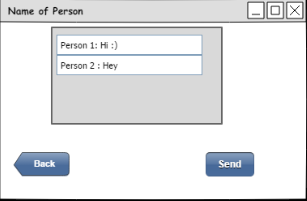
1. Select your about: Dropdown menu listing options for user to choose from.
2. **Enable/Disable Read Receipts**

This screen allows the user to enable or disable read receipts.



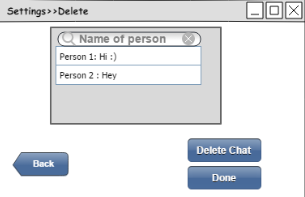
1. **Send Message**

This screen allows two users to share messages, audios, videos and photos.



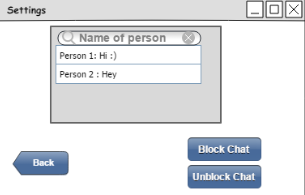
1. **Delete Chat**

This screen lets the user to delete a chat with another user.



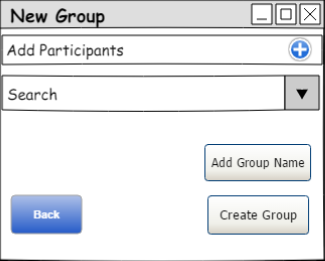
1. **Block/Unblock Chat**

This screen allows the user to block a chat or unblock an already blocked chat.



1. **Create Group**

This screen allows the user to create a group and add participants. The user can also update the name of the group.



The Various fields involved are:

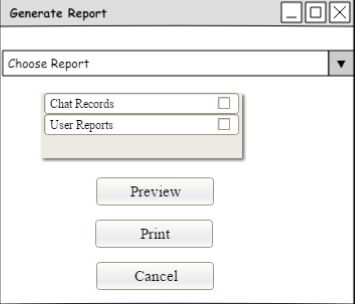
1. Add participant: Dropdown list displays all the contacts of the user.
2. Add Group Name: Alphanumeric of length at max 15 characters. Special characters and emoticons allowed.
3. **Generate Reports**

This screen lets the admin to generate and print reports for all the users.

The following reports can be generated:

1. Chat records
2. User reports

The admin can preview and print the reports generated.



The various fields involved are:

1. Choose report: Dropdown list provides admin options to choose the type of report to generate.
   * 1. **Hardware Interfaces**

As stated in section 2.1.3

* + 1. **Software Interfaces**

As stated in section 2.1.4

* + 1. **Communications Interface**

None

* 1. **Functions** 
     1. **LOGIN**

1. **Use case Description**

|  |
| --- |
| Introduction: This use case allows the user to login to the system. |
| Actors: User, Administrator |
| Pre-condition: The user must have already existing account. |
| Post-condition: The user is successfully logged into the system. |
| Basic flow:   1. The user is required to enter the right credentials i.e. username and password of an already existing account. 2. After the credentials are verified, user is logged into the system. |
| Alternate flow:  Alternate Flow 1: Invalid User Name  If the username is invalid, an error message is displayed and user is asked to enter the details again.  Alternate Flow 2: Invalid Password  If the password is incorrect, an error message is displayed and user is asked to enter the details again.  Alternate Flow 3: User Exits  User cancels the process and exists. |
| Special Requirements: None |
| Associated use cases: None |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore
7. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **ENABLE/DISABLE READ RECEIPTS**

1. **Use case Description**

|  |
| --- |
| Introduction: This use case allows the user to enable/disable the read receipts while chatting. |
| Actors: User |
| Pre-condition: The user must be logged into the system. |
| Post-condition- Read receipts are set according to the user’s preference. |
| Basic flow:   1. The user successfully logs into the system. 2. The user open settings, selects Account and then selects Privacy option. 3. The user enables/disables read receipts in Privacy settings. |
| Alternate flow:  Alternate Flow 1: Invalid Credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: User Exits  User exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore
7. Every user shall be able to change its own Privacy setting for read receipts.
8. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **UPDATE PROFILE PICTURE**

1. **Use case Description**

|  |
| --- |
| Introduction: This use case allows the user to update his/her profile picture. |
| Actors: User |
| Pre-condition: The user must be logged into the system |
| Post-condition: User’s profile picture is successfully updated. |
| Basic Flow:   1. The user successfully logs into the system. 2. The user taps on current display picture and selects Edit. 3. The user browses the image on his system and selects it as the new profile image. |
| Alternate flow:  Alternate Flow 1: Invalid Credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: No image on the system  There is no image on the system he is using.  Alternate Flow 3: User exits  The user exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. User shall be able to update its profile picture if an image exists on the system.
8. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **UPDATE STATUS**

1. **Use Case Description**

|  |
| --- |
| Introduction: This use case allows the user to update his/her status. |
| Actors: User |
| Pre-condition: The user must be logged into the system. |
| Post-condition: User’s status is successfully updated. |
| Basic flow:   1. The user successfully logs into the system. 2. The user selects Settings option and then selects Status. 3. The user enters new status and clicks on Save Changes. |
| Alternate flow:  Alternate Flow 1: Invalid Credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: User exits  The user exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. Every user shall be able to update its status.
8. Status should contain alphanumeric characters and Emoji only.
9. Length of status shall be 30 characters.
10. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **DELETE CHAT**

1. **Use Case Description**

|  |
| --- |
| Introduction: This use case allows the user to delete a chat. |
| Actors: User. |
| Pre-condition: The user must be logged into the system. |
| Post-condition: User successfully deletes a chat. |
| Basic flow:   1. The user successfully logs into the system. 2. The user is allowed to choose a chat to delete. 3. The user opens the settings of the chat and selects the option to delete the chat. |
| Alternate flow:  Alternate Flow 1: Invalid credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: Languages not familiar  The user is not familiar with any of the languages offered by the chat server system  Alternate Flow 3: User exits  The user exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. Only the user is allowed to delete its chat from the system.
8. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **BLOCK/UNBLOCK CHAT**

1. **Use Case Description**

|  |
| --- |
| Introduction: This use case allows the user to block/unblock a chat |
| Actors: User |
| Pre-Condition: The user must be logged on to the system. |
| Post-Condition: Chats are blocked or unblocked according to the user’s preference |
| Basic flow:   1. The user logs onto the system. 2. The user opens the chat to change the settings for. 3. The user opens the settings and chooses to block an unblocked chat or to unblock a blocked chat. |
| Alternate flow:  Alternate Flow 1: Invalid credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: User exits  User exits the system. |
| Special Requirements: None |
| Associated use cases: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. Only the user is allowed to block/unblock a particular user from its contacts.
8. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **SEND MESSAGE**

1. **Use Case Description**

|  |
| --- |
| Introduction: This use case allows the user to send a message to a group or a personal chat |
| Actors: User |
| Pre-Condition: The user must be logged onto the system. |
| Post-Condition: Information is successfully shared. |
| Basic Flow:   1. The user logs onto the system. 2. The user is allowed to send a message on private chat or on group chat. 3. After choosing the chat user gets the option to send a text message, image, audio message or video message. |
| Alternate flow:  Alternate Flow 1: Invalid credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: User exits  User exits the system. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. The user is allowed to send text message, images, audio or video messages.
8. Length of the text message is not limited to a fixed number of characters.
9. The user can send images, audio and video downloaded on system but not through internet.
10. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **GENERATE REPORT**

1. **Use Case Description**

|  |
| --- |
| Introduction: This use allows the administrator to generate reports for chat records and user reports. |
| Actors: Administrator. |
| Pre-Condition: The administrator must be logged onto the system. |
| Post-Condition: Reports are successfully generated. |
| Basic Flow:   1. The administrator logs onto the system. 2. The administrator can generate report on chat records or user reports.   Basic Flow 1: Chat records   1. The admin retrieves data for a chat. 2. The admin generates a report on number of text messages shared, images shared, video and audio messages shared.   Basic Flow 2: User Report   1. The admin retrieves data for the person. 2. The admin generates report for the user for number of messages shared, number of other users blocked by, number of users blocked, total data used. |
| Alternate flow:  Alternate Flow 1: Invalid credentials  Admin enters invalid credentials and is unable to login.  Alternate Flow 2: Admin exits  The admin exits the system. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. Only the administrator is allowed to access the Generate Report module.
8. The administrator shall be able to generate the following reports: -
9. Chat Records
10. User Reports
11. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **GENERATE REPORT**

1. **SEARCH CHAT/USER**

|  |
| --- |
| Introduction: This use case allows the user to search a chat or a user. |
| Actors: User. |
| Pre-Condition: The user must be logged onto the system. |
| Post-Condition: The search query is successfully executed. |
| Basic Flow:   1. The user logs onto the system. 2. The user chooses the search option to search for a chat or a person. 3. The user types the name of the chat and the search results are retrieved. |
| Alternate flow:  Alternate Flow 1: Invalid credentials  User enters invalid credentials and is unable to login.  Alternate Flow 2: Language not familiar  The user is not familiar with any of the languages offered by the chat server system.  Alternate Flow 3: User exits  The user exits the system. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. The user is allowed to search another user by his name saved in system.
8. The user shall be allowed to search a chat by using appropriate search keywords.
9. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **MAINTAIN GROUP DETAILS**

1. **USECASE DESCRIPTION**

|  |
| --- |
| Introduction:This use case documents the steps that the user must follow in order to maintain group details and add, update, delete and view group information. |
| Actors**:** User |
| Pre-Condition:The user must be logged onto the system before this use case begins. |
| Post Condition: If the use case is successful, then the group information is added, updated, deleted or viewed. If not, the system state is unchanged. |
| Flow of Events  Basic Flow   1. This use case starts when user creates a group and wishes to add/delete/update/view group information. 2. The system requests that the user specify the function he/she would like to perform (add user, remove user, update group details, view group details).   Basic Flow 1: Add Participant   1. The system requests the user to enter user information. This includes: 2. Contact No. 3. Name 4. The new participant is then added to the group.   Basic Flow 2: Delete Participant   1. The system requests the user to enter user information. This includes: 2. Contact No. 3. Name 4. The corresponding participant user is deleted from the group.   Basic Flow 3: Update Group Information   1. The system requests the user to enter group information. This includes: 2. Group Name 3. No. of participants 4. Name of all participants 5. Admin Rights 6. The user makes the desired changes to group information. 7. The system updates group information with updated information   Basic Flow 4: View Group Information   1. The system requests the user to enter group information. This includes: 2. Group Name 3. No. of participants 4. Name of all participants 5. Admin Rights 6. The system retrieves the information and displays it to the user. |
| Alternate Flow  Alternate Flow 1: Invalid Login Details  If in the basic flow, the user enters an invalid login id, password the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the login at which point the Use Case ends.  Alternate Flow 2: Participant Already Exists  If in Add Participant flow, a participant with same name and contact no. already exists, the system displays an error message. The user returns to basic flow and may re-enter user information.  Alternate Flow 3: User not found  If in Add Participant flow, a participant with specified name and contact no. does not exist, the system displays and error message. The user returns to basic flow and may re-enter user information.  Alternate Flow 4: Update Cancelled  If in Update Group Information flow, user decides not to update the information, update is cancelled and basic flow is restarted at beginning.  Alternate Flow 5: Deletion Cancelled  If in Delete Group Information flow, user decides not to delete the participant, delete is cancelled and basic flow is restarted at beginning.  Alternate Flow 6: User Exits  This allows the user to exit at any time during the use case. The use case ends. |
| Special Requirement None |
| Associated Use Case: Login |

1. **Validity Checks**
2. Login Id cannot be left blank.
3. Password cannot be blank
4. Login Id can have alphanumeric characters.
5. Password will accept 4-15 characters
6. Password can accept hyphen and underscore.
7. User name cannot be left blank.
8. User name shall contain only alphabetic characters and blank spaces.
9. Length of user name can be of 3 to 50 characters.
10. Contact No. cannot be left blank.
11. Every user has a unique contact number.
12. Contact no. can be up to 11 digits.
13. Contact no. shall accept only numeric digits.
14. Group Name cannot be left blank.
15. Group Name can have alphanumeric characters.
16. Length of group name can be up to 50 characters.
17. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

* + 1. **MAINTAIN USER DETAILS**

1. **USECASE DESCRIPTION**

|  |
| --- |
| Introduction: This use case enables the administrator to maintain user details. |
| Actors: Administrator. |
| Pre-Condition: The administrator must be logged onto the system. |
| Post-Condition: The user maintains are successfully maintained. |
| Basic Flow:   1. The administrator logs onto the system. 2. The system requests the administrator specify the function he/she would like to perform (either add a user, update a user, delete a user or view a user). 3. Once the administrator provides the requested information, one of the following sub flows executes.   Basic Flow 1: Add a User   1. The administrator checks for the uniqueness of the user name and the constraints for the password. 2. The user is added to the database and his/her details are maintained and reports are generated.   Basic Flow 2: Update a user   1. The administrator enters the name of the user. 2. The system displays the user information. 3. The administrator makes the required changes.   Basic Flow 3: Delete a user   1. The system requests the administrator to specify the user name. 2. The system displays the information of the user. 3. The system prompts the administrator to confirm the deletion of the user. 4. The administrator verifies the deletion. 5. The system deletes the record.   Basic Flow 4: View a user   1. The system requests the administrator to enter the user name. 2. The system retrieves and displays the user information. |
| Alternate Flow:  Alternate Flow 1: Invalid Entry  If the user name entered does not exist the system displays an appropriate error message. The actor returns to the basic flow and may re-enter the invalid entry.  Alternate Flow 2: User name already exists  If in Add a user flow, a user name already exists then the system displays an error message. The administrator returns to the basic flow and may re-enter the user name.  Alternate Flow 3: User name not found  If in Delete, update or view flow, the user name does not exist, the system displays an error message. The administrator returns to the basic flow and re-enters the user name.  Alternate Flow 4: Update cancelled  If in update a user, the administrator decides not to update a user details, update is cancelled and the administrator returns to the basic flow.  Alternate Flow 5: Delete cancelled  If in update a user, the administrator decides not to delete a user details, delete is cancelled and the administrator returns to the basic flow.  Alternate Flow 6: User exits  This allows the actor to exit the at any time during the use case. The use case ends. |
| Special Requirements: None |
| Associated Use Case: Login |

1. **Validity Checks**
2. Only, the administrator is allowed to access the ‘Maintain User Details’ module.
3. Login Id cannot be left blank.
4. Password cannot be blank
5. Login Id can have alphanumeric characters.
6. Password will accept 4-15 characters
7. Password can accept hyphen and underscore.
8. User name cannot be left blank.
9. User name shall contain only alphabetic characters and blank spaces.
10. Length of user name can be of 3 to 50 characters.
11. Contact No. cannot be left blank.
12. Every user has a unique contact number.
13. Contact no. can be up to 11 digits.
14. Contact no. shall accept only numeric digits.
15. **Sequencing Information**

None

1. **Error Handling/Response to abnormal situations**

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

**3.3 Performance requirements**

|  |
| --- |
| 1. Should support at least 50 users simultaneously. 2. Should run on 2.50GHz processor, 2.00 GB RAM server. 3. Responses should be within 2 seconds. |

**3.4 Logical Database requirements**

The following information will be placed in a database:

|  |  |
| --- | --- |
| **Table name** | **Description** |
| Login | Records the login details of the user |
| User | Records the details of the User |
| Group | Records the details of each group i.e. group admin, group members and group messages. |
| Chat | Records the details of chat between users. |
| Message | Records the details of message sent between users. |

**3.5 Design Constraints**

None

* 1. **Software System Attributes**

**Usability**

The proposed chat server system will be user friendly and easy to operate and the function will be easily understandable.

**Reliability**

The system will be available to the users through the registration period and have a high degree of fault tolerance.

**Security**

The system will be password protected. Users will have to enter correct login ID and password to access the application  
**Maintainability**

The system will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual module

**Portability**

The chat server system application will be easily portable on any windows based system that has SQL Server installed.

**Efficiency**

The system application will cater to the requirements of the user and will be highly efficient in terms of speed and response time.

**3.7 Other Requirements**

None

**EXPERIMENT- 6**

**AIM:** To prepare Class Diagram for Chat Server System.

**THEORY:**  In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

* The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
* The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
* The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.
* A class with three compartments.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

**RESULT:**

**DISCUSSIONS:** We learnt to draw the Class diagram of the “Chat Server System” case study by identifying use cases, Actors and relationships between them.

**CONCLUSION:** Class diagram Descriptions for Chat server system has been implemented.

**EXPERIMENT - 7**

**AIM:** To prepare Sequence Diagram for Chat Server System.

**THEORY:** A sequence diagram is an [interaction diagram](https://en.wikipedia.org/wiki/Interaction_diagram) that shows how objects operate with one another and in what order. It is a construct of a [message sequence chart](https://en.wikipedia.org/wiki/Message_sequence_chart).

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realisations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

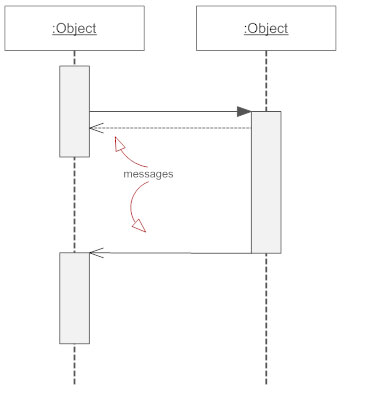
Class Roles or Participants  
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.



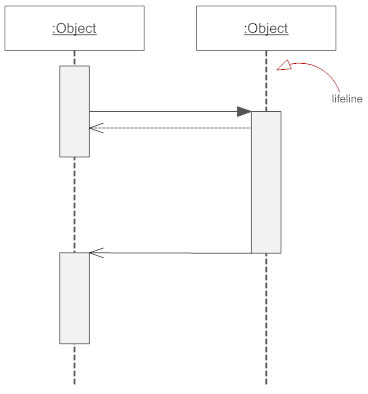
Activation or Execution Occurrence  
Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.



Messages  
Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.



Lifelines  
Lifelines are vertical dashed lines that indicate the object's presence over time.



Destroying Objects  
Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

**RESULT:**

Sequence Diagram

* Login

1. Basic Flow 1



1. Alternate Flow 1- Invalid Details



* Update Profile Picture

1. Basic Flow 1



1. Alternate Flow 1- No Image On System



* Update Status

1. Basic Flow 1



* Delete Chat

1. Basic Flow 1



* Enable/Disable Read Receipts

1. Basic Flow 1- Enable Read Receipts



* Search Chat/User

1. Basic Flow 1- Search Chat



* Block Chat

1. Basic Flow 1



* Send Message

1. Basic Flow 1



* Generate Report

1. User Report



1. Chat Report



* Maintain Group Details

1. Basic Flow 1- Add Participant

****

1. Basic Flow 2- Delete Participant

****

1. Basic Flow 3- Update Group Details

****

1. Basic Flow 4- View Group Details

****

1. Alternate Flow 1- Participant Already Exists

****

1. Alternate Flow 2- User Not Found

****

1. Alternate Flow 3- Update Cancelled

****

1. Alternate Flow 4- Delete Cancelled

****

* Maintain User Details

1. Basic Flow 1- Add User

****

1. Basic Flow 2- Update User

****

1. Basic Flow 3- Delete User

****

1. Basic Flow 4- View User

****

1. Alternate Flow 1- Invalid Entry

****

1. Alternate Flow 2- User Name Already Exists

****

1. Alternate Flow 3- User Name Not Found

****

1. Alternate Flow 4- Update Cancelled

****

1. Alternate Flow 5- Delete Cancelled

****

**DISCUSSIONS:** We learnt to draw the Sequence diagram of the “Chat Server System” case study by identifying use cases, Actors and relationships and messages transferred between them.

**CONCLUSION:** Sequence diagram Descriptions for Chat server system has been implemented.

**EXPERIMENT- 8**

**AIM:** To prepare Collaboration Diagram for Chat Server System.

**THEORY:** A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). The concept is more than a decade old although it has been refined as modelling paradigms have evolved.

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects. As the number of objects and messages grows, a collaboration diagram can become difficult to read. Several vendors offer software for creating and editing collaboration diagrams.

**RESULT:**

Collaboration Diagram

* Login

1. Basic Flow 1



1. Alternate Flow 1- Invalid Details



* Update Profile Picture

1. Basic Flow 1



1. Alternate Flow 1- No Image On System



* Update Status

1. Basic Flow 1



* Delete Chat

1. Basic Flow 1



* Enable/Disable Read Receipts

1. Basic Flow 1- Enable Read Receipts



* Search Chat/User

1. Basic Flow 1- Search Chat



* Block Chat

1. Basic Flow 1



* Send Message

1. Basic Flow 1



* Generate Report

1. User Report



1. Chat Report



* Maintain Group Details

1. Basic Flow 1- Add Participant

****

1. Basic Flow 2- Delete Participant

****

1. Basic Flow 3- Update Group Details

****

1. Basic Flow 4- View Group Details

****

1. Alternate Flow 1- Participant Already Exists

****

1. Alternate Flow 2- User Not Found

****

1. Alternate Flow 3- Update Cancelled

****

1. Alternate Flow 4- Delete Cancelled

****

* Maintain User Details

1. Basic Flow 1- Add User

****

1. Basic Flow 2- Update User

****

1. Basic Flow 3- Delete User

****

1. Basic Flow 4- View User

****

1. Alternate Flow 1- Invalid Entry



1. Alternate Flow 2- User Name Already Exists



1. Alternate Flow 3- User Name Not Found



1. Alternate Flow 4- Update Cancelled



1. Alternate Flow 5- Delete Cancelled



**DISCUSSIONS:** We learnt to draw the Collaboration diagram of the “Chat Server System” case study by identifying use cases, Actors and relationships and messages transferred between them.

**CONCLUSION:** Collaboration diagram Descriptions for Chat server system has been implemented.

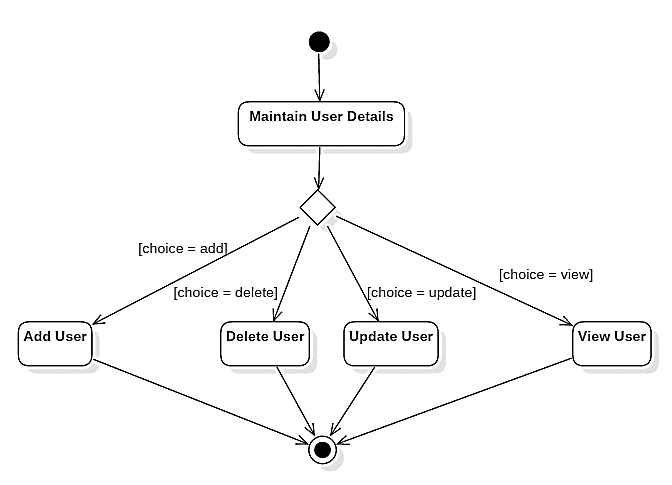
**EXPERIMENT 9**

### AIM: To draw the activity diagram for the Chat Management System project.

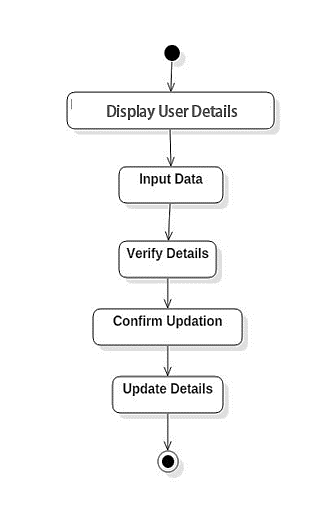
**THEORY:** An Activity diagram is a visual representation of any system's activities and flows of data or decisions between activities. They provide a very broad view of a business process. They represent the dynamics of a system. These are flow charts that are used to show the work flow of a system. They show what activities can be done in parallel, and any alternative paths through the flow.

**RESULT:**

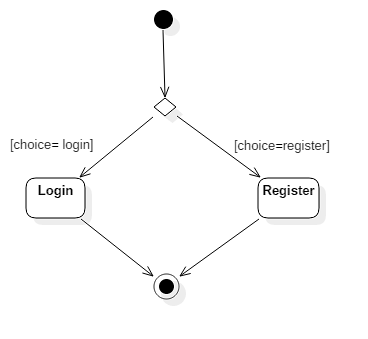
**MAINTAIN USER DETAILS**



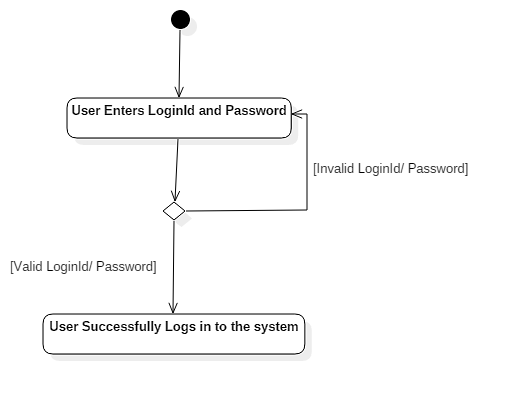
**UPDATE USER DETAILS**



**LOGIN/REGISTER**



**LOGIN**



**BLOCK/UNBLOCK CHAT**



**DISCUSSION:** From this practical we have learnt how to identify flow of control from activity to activity in the application and draw the Activity diagram.

**CONCLUSION:** Activity diagram for Chat server system has been implemented.