**Introduction**

**Purpose**

The purpose of this document is to layout a detailed description of the functionality of the messaging application {product name}.

**Intended Audience and Reading Suggestions (More to be added)**

* **Businesses?**

**Product Scope**

{product name} is a messaging application that allows people to message each other individually, or through groups effectively and securely. Users will be capable of easily retrieving and initiating conversations all while having the information of their communication partners readily available.

**Overall Description**

|  |  |  |
| --- | --- | --- |
| **Group** | **Requirement** | **Priority** |
| Login | The system must support unique user log in information. A user will register a username and password, which the system will store and keep as persistent data. | High |
| Login | The system will support user login from third party sites, such as husky, LinkedIn, and Facebook. | Medium |
| Login | The system may support password restrictions. This could include capitalization, unique character requirements, or minimum character length. | Low |
| Login | The system will encrypt the user’s login information when they create a new account and will store it encrypted while at rest. | High |
| Login | The system will encrypt a user’s login information when they are attempting to login and will query the encrypted data against the database. The system will never see the user login information unencrypted. | High |
| Messaging | Users must have the ability to send messages to channels. This will include private messages, group channels, feeds, etc. The user will create a new message client side, and then will send the message to the server. | High |
| Messaging | All messages must be sent to a specific channel. | High |
| Messaging | When a user sends a message to a channel, the system will push the message to all active users in that channel. | High |
| Messaging | The system will store all messages that a user sends to it. Stored messages will be able to be recalled later. | High |
| Messaging | The system will encrypt all messages received from clients and transmit them encrypted to the receiving clients. | Medium |
| Messaging | Messages will be able to support MIME types. | Medium |
| Messaging | The system will support message threading. Threading a message will allow users within a channel to reply to other messages already present in that channel. The threaded message will appear below the main message and will read similar to how email chains read. | Medium |
| Messaging | The system will support message recalling. Recalling a message will delete it from the user’s queue. A message can only be recalled if the user has not read it yet. Only messages sent to a user’s private channel can be recalled. | Medium |
| Messaging | The system may provide the user functionality to queue up messages to be sent at a specific time. Queued messages will be sent to the server but will not be sent from the server to the respective clients until the allotted time has passed. | Low |
| Messaging | Messages may have a lifetime associated with them. After their lifetime has expired, they will no longer be displayed to the clients. Expired messages will still be stored by the system | Medium |
| Messaging | All messages will have a timestamp. | Medium |
| Messaging | Messages will be delivered/rendered in the order they were sent. This is determined by the timestamp assigned to the message. | High |
| Messaging | The system may support hashtags within messages. Hashtags allow users to search for any messages containing specific hashtags. | Low |
| Channel | Server treats channels as a function of groups | High |
| Channel | Messages travel through channels to specific channelIDs that represent either users or groups. | High |
| Channel | Server-side implementation of channel interface will be implemented via group-wide channel, DM channel and wall channel | High |
| Search | Users should be capable of searching for other users | High |
| Search | Users should be capable of searching for messages in channels | High |
| Search | Users should be capable of searching to find other channels by name | High |
| Events | Users should be capable of scheduling meetings and events | Medium |
| Member | Users should be able to use “personality” icons for their profiles. | Low |
| Member | Users should be able to set availability (Do not Disturb, Online, Unavailable) | Low |
| Member | Users should be able to set their profile to private, unable to be searched for. | Low |
| Groups | Users can be part of groups. | High |
| Groups | Groups must have at least one user, and at least one moderator | High |
| Groups | Moderators can kill groups, and can dictate who can be invited and rejected from joining | High |
| Groups | Groups can have more than one moderator. | High |
| Groups | Moderators can transfer powers to other users in a group. | High |
| Groups | Groups can be protected by passwords. This functionality can be controlled by the moderator/moderators | High |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

:

**Potentially can add more info:**

* **Product Perspective**
* **Design and implementation Constraints**
* **User Documentation**
* **Assumptions and Dependencies**
* **UI Requirements**
* **Hardware Interfaces**
* **Software Interfaces**
* **Infrastructure (AWS or Azure)**
* Elastic Beanstalk autoscaling?
* EC2 …
* S3 Storage (Encryption at rest) …
* DynamoDB or SQL Database …
* Amazon Cognito - Login

**Nonfunctional Requirements**

**Systems Scalability**

* Application should be capable of scaling to a few thousand users

**Security Requirements**

* Messages should be capable of having end to end encryption (Tier offering)
* Messages should be encrypted at rest
* Private messages should have copy/paste and forwarding functionality removed

**Calea**

* Government should have the ability to obtain specific user data both historical and in real time

Messages may need to be wrapped with from and to IP addresses

* All employee changes and viewings of objects should be recorded and stored
* We do not need to decrypt anything we get that is encrypted

**Non User Classes and Characteristics (Tier System )**

* **Free: TBD**
* **Paid: TBD**

**Reach Functionality**

**Operating Environment**

* Application should have the ability to be accessed regardless of device.

**GrabBag**

* Systems may notify users of screen captures
* Parental Control: Messages may be capable of being reported/flagged by users
* Messages may be capable of translation to receivers
* Messages may offer a filtering option
* Users may have the ability to RSVP to events

**Glossary**

**TBD**