CONFIDE Instant Messaging Service

Vision

# Introduction

Pinnacle Pty Ltd have commissioned Team Orange to develop an instant messaging platform “Confide” for corporate use with their company. Pinnacle is an investment company which specialises in providing management of superannuation products for high net worth individuals. They want to be able to react quickly to changes in the market and share information among their staff as it comes to hand. At the same time, Pinnacle’s management are highly sensitive to their intellectual property and maintaining their track record of integrity and protection of customer’s sensitive information. Having a messaging application unique to the company is symbolic of these goals.

For these reasons, Pinnacle wish to develop a privately hosted instant messaging client-server application. The company has requested full control of and rights to the source code. The messaging client will be developed in accordance with Pinnacles specific requirements.

# Positioning

## Problem Statement

The problem is most commercially available instant messaging systems have one or more of the following (undesirable) characteristics:

* Proprietary code base
* Public hosting
* Open source, but not tailored to specific needs

This rules out most of the offerings currently available to Pinnacle.

A privately developed solution will be designed to run on Pinnacle’s own hardware. It will have only the features that Pinnacle wants and Pinnacle will own the rights to this software if they want to make changes or improvements to its functionality in the future.

## Product Position Statement

For Pinnacle, who are seeking to provide their staff with an instant messaging platform, the Confide custom instant messaging client combined with best-of-breed messaging server and protocol, will deliver the corporate identity, security and customisation that only an in-house product can deliver.

Commercial Off the Shelf (COTS) products and most open source alternatives must aim to support the widest variety of use cases. There is no individual group which can dictate the direction of the roadmap. Public access to the source code increases the possibility of vulnerabilities being found and used in exploits against Pinnacle. By developing the source code privately this risk is reduced.

# Stakeholder Descriptions

## Stakeholder Summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| Pinnacle staff | Day to day tasks within the company | Main users of the application |
| IT Support | Maintain Pinnacle IT infrastructure | This will become a supported application. |
| Management | Seeking to provide a productive, harmonious environment within the company. Meeting the needs of shareholders. | Main sponsor of the product. Supervise rollout and adoption as the endorsed communication product. As Pinnacle staff, management will also be using the product. |
| Developers | Enhance and maintain product functionality | Manage product improvements and fix issues raised by IT Support and the users |

## User Environment

Initially the product will be a desktop application. Usage will be via corporate laptop and desktop computers. Pinnacle Management envisages that a well-designed instant messaging product will help to promote the corporate identity while still maintaining security and productiveness.

# Product Overview

## Needs and Features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Business Requirement (Need)** | **Priority** | **Requirement Reference** | **Planned Release** |
| 1 | Create an account | High | Authentication  Authorisation |  |
| 2 | Edit account | High | Edit account personal details  Change picture/avatar  Set presence |  |
| 3 | Add user to personal contacts | High | Search for user  Accept deny request |  |
| 4 | View contacts | High | Search contacts  View contact’s presence  View additional contact data |  |
| 5 | Manage Contacts | High | Block/Delete/Mute contact |  |
| 6 | Start a chat | High | Send/receive messages  Notification of replies  New chat or existing thread |  |
| 7 | Multi-user chat | Medium | Initiate/Join Multi-user chat  Notification to join |  |
| 8 | View chats | High | Search for and display chat threads |  |
| 9 | Attach file to chat | Medium | Add document/image/video |  |
| 10 | Secure chat | Low | End to end encryption of chat |  |
| 11 | Employees can be in any location on the Internet and connect to other Employees. |  |  |  |
| 12 | Scalable and flexible to support future growth and requirements of Business. |  |  |  |
| 13 | Supports availability requirements of Business. |  |  |  |
| 14 | Security and Auditing Policies can be applied to the Service. |  |  |  |

# Other Product Requirements Confide chat service will operate as one of a package of corporate communication methods at Pinnacle (including voice, video conferencing and email). The service is described as a “must-have” but minor failures of the service will not cause major disruption to the business due to the alternative communication methods at their disposal.

Pinnacle have already moved their infrastructure to data centres which are compliant with the local standards and codes. These data centres also provide 99.99% guaranteed uptime for the network and hardware, with build in redundancy measures. Pinnacle have also purchased data replication and backup services.

Pinnacle make use of virtualisation technologies. This allows flexibility to configure the hardware and operating system specifications to suit the project on the server side. No significant challenges are anticipated for provisioning the correct hardware for production use (server side).

On the client side, initial discussions have lead to the recommendation that development should be in a managed language such as c# or Java. This simplifies the deployment to the range of operating systems currently supported at Pinnacle.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
|  |  |  |

# Functional Requirements

## Usability Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Success Criteria** | **Priority** |
| 1.1 | User Interface must have functional work flow that is intuitive | Users with minimal computer levels are able to use the with little training. | High |
| 1.2 | User Interface must provide simple and effective experience | Users are able to perform business communication efficiently. | High |
| 1.3 | Users can submit messages if no Users are present. | Message are stored within Channel. | High |
| 1.5 | User can submit drafted messages at will. | Message are only sent to Channel after User's request to Send. | High |
| 1.6 | User can send request to access Channel. | Notification of Channel Request are received by other Users. | Low |
| 1.7 | User can accept or refuse access to Channel | A dialogue is created to accept or deny request. | Low |
| 1.8 | User settings are restored. | Settings are automatically initialized when application opens. | Low |

## Performance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Success Criteria** | **Priority** |
| 2.1 | Multiple Users can be added to Channels. | Channels can exist with Multiple Users. | High |
| 2.2 | The confidentiality of Messaging Data is ensured between the transit User devices. | Users messages can not be read by unknown actors. | Medium |
| 2.4 | The Service is reliable and has acceptable response times to Users. | Messages and presence are communicated in effective time frame. | Medium |
| 2.5 | Can be installed on popular OS. | Users can install Application on Multiple types of OS | Medium |

## 

## Administrative Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Success Criteria** | **Priority** |
| 3.1 | User can create a new account on the Service. | Users can submit a request for a new account and submit a username and password. | High |
| 3.2 | Users will be require username and password to access Service. | A username and password is summit during a logon. | High |
| 3.3 | User can create a new Channel. | Channel can be created by Users and assign to the Users account. | High |
| 3.4 | User can delete Channel. | Channels can be removed by the Users. | Medium |
| 3.7 | User can block other User. | Blocked User cannot send messages or Channel requests to User. | Low |

# 

## Interface Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Success Criteria** | **Priority** |
| 4.1 | Login & logout of Service. | An Interface is available to manage Login & logout process. | High |
| 4.2 | Read messages In Channel | View all Channel Messages in human readable and time categorised format. | High |
| 4.3 | Switch between Channels in view. | Option to switch between Channels Messages in view. | Medium |
| 4.4 | Assign Users to Channel. | Options to assign Users to Channel | High |
| 4.5 | Users presence displayed in Channel. | Indicates the presence of a User. | High |
| 4.6 | Users who have access to Channel will be displayed. | A list of assigned Channels | High |
| 4.7 | Submit drafted messages at will. | A text box of the drafted message content. | High |
| 4.8 | State of connection to Service is displayed. | Visual indication of state. | Medium |
| 4.9 | Notified of new messages in Channel. | Visual and audible indication. | Medium |
| 4.10 | Notified of new Channel request. | Visual and audible indication. | Low |

# 

# Non-Functional Requirements

## Administrative Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Measurement** | **Priority** |
| 1.1 | Service supports multiple Users. | Multiple Users can be assigned to Channels. | High |
| 1.2 | Channels will achieve messages for post-time access. | Messages are stored on database and can be accessed by channel with Time-Stamp. | High |
| 1.3 | Data will be stored on centralized infrastructure. | Information is saved and consistent among multiple uses and users. | Low |
| 1.4 | Users will be authenticated to access Channels. | Users can only access their domain once credentials have be verified. | High |
| 1.5 | User can authenticate other Users to access Channel. | Users can only access non-subscribed Channel once authorised by Owner. | High |
| 1.6 | User can de-authenticate other Users to access Channel. | Users can unsubscribe User from Channel. | High |
| 1.7 | Users will be authenticated upon logging in. | A user's login information (username and password) must be known in order to log in to that user's account. | High |
| 1.8 | Accounts will be authorized as either regular user or administrator. | It is possible for both regular user and administrator accounts to be created. And only administrators are capable of creating administrator accounts. Administrator accounts will have a higher level or authorization. | High |

## User Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Measurement** | **Priority** |
| 2.1 | User configuration data saved on own machine. | Settings configuration is initialized during Application opening, Data is RW onto host machine. | High |
| 2.2 | User devices can run on Windows, OS X, and Linux environments. | Service can be deployed on Windows, OS X, and Linux Environments. | Low |
| 2.3 | User devices can access Server from typical Internet enabled devices. | Client devices are able to estable connection to Server through non-heterogeneous networks. | High |
| 2.4 | Usable for average people, without training necessary. | Average people, who are used to computers and desktop applications, can use the client application without any training. | Medium |
| 2.5 | Usable for color blind people. | People with all common kinds of color blindness can still easily use the application. | High |
| 2.6 | The Service is maintainable and extendible. | The underlying communication protocols are an open or international standard. | Medium |
| 2.7 | User devices can detect we other Users are online | The underlying communication protocol notifies devices of active users. | High |

## Quality Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Measurement** | **Priority** |
| 3.1 | The User side components are maintainable and Extendable. | The client side components are architecturally neutral, built in a well structured and designed object oriented way. | High |
| 3.2 | Messages are exchanged between devices within 1 second. | Message exchange has statistical average under 1 second (assuming decent internet connection on all client sides). | Low |
| 3.3 | The service side components are maintainable and Extendable. | The server side components are architecturally neutral, well supported and adaptable to new functionally. | High |
| 3.4 | Users Client devices can be connected to Server through most traditional networks. | Client devices can be connected through non-heterogeneous networks. | Medium |
| 3.5 | High speed and quality development of client side application. | The programming language used is known well by all team members, high level, object oriented. (Java) The client side API allows pure Java to be used. | Medium |
| 3.6 | Risks avoided by using tried and tested components. | All components used are mature, meaning they are tried, tested, and stable. |  |

## Security Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Measurement** | **Priority** |
| 4.1 | Service can support P2P Encryption. | Data Confidentiality is supported by the use of the TLS Protocol as a wrapper around messaging data. | Medium |
| 4.2 | Security policy can be strongly implemented and enforced. | Security Policy can be globally applied to User base. | Low |
| 4.3 | Auditing policy can be strongly implemented and enforced. | Auditing Policies can be globally applicated to User base. | High |
| 4.4 | Service can be extended to support external backed up on another database. | External database contains up to date extra backup of the data that is stored on the server database. | very low |

# 

# 

// lost password retrieval

# System Constraints

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Constraint** | **Measurement** | **Priority** |
| 1 | Server that supports XMPP |  | High |
| 2 | Client side implementation language will be Java |  | High |
| 3 | System must be completed by 16/10/17 |  | Very High |
| 4 | The system shall be available 99.9% of the time |  | High |
| 5 | The client side application myst work on Windows, OS X, and Linux. |  | High |

# External Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Interface** | **Measurement** | **Priority** |
| 1 | Openfire server |  | High |
| 2 | Smack API |  |  |
| 3 |  |  |  |

# 