Team Orange

CONFIDE Instant Messaging Service Vision

1. Table of Contents

1. Introduction 3

2. Positioning 3

 2.1 Problem Statement 3

 2.2 Product Position Statement 3

3. Stakeholder Descriptions 4

 3.1 Stakeholder Summary 4

 3.2 User Environment 4

4. Product Overview 4

 4.1 Needs and Features 4

5. Other Product Requirements 5

6. Functional Requirements 6

 6.1 Usability Requirements 6

 6.2 Performance Requirements 6

 6.3 Administrative Requirements 7

 6.4 Interface Requirements 7

7. Non-Functional Requirements 8

 7.1 Administrative Requirements 8

 7.2 User Requirements 8

 7.3 Quality Requirements 9

 7.4 Security Requirements 10

8. System Constraints 10

9. External Interfaces 10

2. 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 Pinnacle's specific requirements.

3. Positioning

3.1 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 or privately co-located data centre. 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.

3.2 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.

4. Stakeholder Descriptions

4.1 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

4.2 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 productivity.

5. Product Overview

5.1 Needs and Features

Use Case ID[1]	Business Requirement (Need)	Priority	Requirement Reference	Planned Release
UC0001	Create an account	High	Authentication Authorisation	4th Qtr. 2017
UC0002	Add user to personal contacts	High	Search for user Accept/deny request	4th Qtr. 2017
UC0012	Edit account	High	Edit account personal details Change picture/avatar Set presence	4th Qtr. 2017
UC0004	View contacts	High	Search contacts View contact's presence View additional contact data	4th Qtr. 2017

UC0002	Manage Contacts	High	Block/Delete/Mute contact	4th Qtr. 2017
UC0007	Start a chat	High	Send/receive messages Notification of replies New chat or existing thread	4th Qtr. 2017
UC0013	Multi-user chat	Medium	Initiate/Join Multi-user chat Notification to join	4th Qtr. 2017
UC0005	View chats	High	Search for and display chat threads	4th Qtr. 2017
UC0006	Secure chat	Low	End to end encryption of chat	4th Qtr. 2017

Note: 1. Refer to RequirementModel.docx for further details of the use cases.

6. 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 led to the recommendation that development should be in a managed language such as Java. This simplifies the deployment to the range of operating systems currently supported at Pinnacle.

7. Functional Requirements

7.1 Chat

ID	Requirement	Success Criteria	Outcome	Priority	Status
1.3	Users can submit messages if no other users are currently online.	Message are stored on server.	Messages can be sent to user, the sent messages are retrieved on the next login.	High	Achieved
1.5	User can submit drafted messages at will.	Message are only sent to contact after user clicks "Send"	Messages can be drafted in text box and are only sent on Enter Key or Send Button.	High	Achieved
1.6	User can send request to access Multi-User chat.	Notification of Multi-User Chat Request are received by other Users.	A notification can be sent to a group of users in contact list.	Low	Achieved
1.7	User can accept or refuse access to Multi-User chat	A dialogue is created to accept or deny request.	After MUC request a dialogue box is displayed in the MUC list.	Low	Achieved
2.1	Multiple Users can be added to Multi-User chats.	Multi-User chats can exist with Multiple Users.	Users can be notified to join chat after creation.	High	Achieved

7.2 Administrative

ID	Requirement	Success Criteria	Outcome	Priority	Status
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.	The account screen has UI features to create an account.	High	Achieved
3.2	Users will require username and password to access Service.	A username and password is summit during a logon.	A username a password is required on the account page to gain access.	High	Achieved
3.3	User can create a new Multi-User chat.	Multi-User chat can be created by Users and assign to the Users account.	A user can create a new MUC from the MUC screen.	Medium	Achieved
3.4	User can delete Multi-User chat.	Multi-User chats can be removed by the Users.	A user can delete the MUC from the MUC list view screen.	Medium	Achieved
3.7	User can block another User.	Blocked User cannot send messages to User.	A user can be blocked from the user list in the chat screen.	Low	Achieved

7.3 Workflow

ID	Requirement	Success Criteria	Outcome	Priority	Status
4.1	Login & logout of Service.	An Interface is available to manage Login & logout process.	An account screen is provided to login to service, a logout button is available after login.	High	Achieved
4.2	Read messages In Chats	View all Chat Messages in human readable and time categorised format.	Messages are displayed in a list view for each chat.	High	Achieved
4.3	Switch between Chats in view.	Option to switch between Chat Messages in view.	A navigation pane is alway present to switch between views.	Medium	Achieved
4.4	Assign Users to Multi-User chat.	Options to assign Users to Multi-User chat	Users are instead notified to join chat.	Low	Removed
4.5	Users presence displayed in Chat.	Indicates the presence of a User.	Presence of the user is displayed in the chat screen list by a red cross and green tick.	High	Achieved
4.6	Users who have access to Multi-User chat will be displayed.	A list of subscribed users.	MUC Chat Screen UI were not extended to allow list of users, components are available to implement.	High	Not Complete
4.8	State of connection to Service is displayed.	Visual indication of state.	Connection status is always displayed on status bar on top screen.	Medium	Achieved
4.9	Notified of new messages in chat.	Visual and audible indication.	A new message notified in MUC/Chat list displays. A new message icon is used. Audible notification	Medium	50% Achieved

			not implemented.		
4.10	Notified of new Chat Request.	Visual and audible indication.	A new message notified in MUC/Chat list displays. A new message icon is used. Audible notification not implemented.	Low	50% Achieved

8. Non-Functional Requirement

8.1 Usability

ID	Requirement	Success Criteria	Outcome	Priority	Status
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.	Beta testing demonstrated users could effectively navigate workflow.	High	Achieved
1.2	User Interface must provide simple and effective experience	Users are able to perform business communication efficiently.	Beta testing demonstrated app UI experience is intuitive and effective for communication.	High	Achieved

8.2 Maintainability

ID	Requirement	Success Criteria	Outcome	Priority	Status
3.1	The client side components are maintainable and extensible.	The client side components are architecturally neutral, built in a well-structured and designed object-oriented way.	Maintainability achieved with MVC Controller pattern, pure Java implementation and open source/IEEE standard protocols used.	High	Achieved
3.3	The service side components are maintainable and Extensible.	The server side components are architecturally neutral, well supported and adaptable to new functionally.	The Service uses Openfire XMPP Server Components. Services are able to integrate to other XMPP compliant Services.	High	Achieved
3.7	User devices can run on Windows environment and possible extended support for OS X, and Linux environments.	Service can be deployed on Windows, OS X, and Linux desktop Environments.	Application can be run on multiple environments with pure java implementation. Project was descoped to windows desktop, primary testing on windows.	Low	Achieved

8.3 Security

ID	Requirement	Success Criteria	Outcome	Priority	Status
4.1	Service can support P2P Encryption.	Data Confidentiality is supported by the use of the TLS Protocol as a wrapper around messaging data.	XMPP components will support this although it was not integrated into app..	Low	Not Achieved
4.2	Security policy can be strongly implemented and enforced.	Security Policy can be globally applied to User base.	User Authentication, Blocking, Monitoring, Auditing can be achieved through server admin console.	Low	Achieved
4.3	Auditing policy can be strongly implemented and enforced.	Auditing Policies can be globally applicated to User base.	Logs are kept of all user activity through server admin console.	Low	Achieved
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.	Services is extendable to all modern database thus supports external backup.	very low	Achieved
4.5	Users will be authenticated to access Chats.	Users can only access their domain once credentials have been verified.	Feature to authenticate chat achieved.	High	Achieved
4.6	User can authenticate other Users to access Multi-User chat.	Users can only access non-subscribed Multi-User chat once authorised by Owner.	Feature to request MUC room achieved.	Low	Achieved
4.7	User can unsubscribe other Users to access chat.	Users can unsubscribe other user from chat.	Feature to Block Contacts achieved.	Low	Achieved
4.8	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.	XMPP Client Login Integrated.	High	Achieved

8.4 Performance

5.1	Messages are exchanged between devices within 1 second.	Message exchange has statistical average under 1 second (assuming decent internet connection on all client sides).	Service speeds meets jitter time.	Low	Achieved
5.2	Users Client devices can be connected to Server through most traditional networks.	Client devices can be connected through non-heterogeneous networks.	Applications can work through multiple domains and NAT.	Medium	Achieved
5.3	Data will be stored on centralised infrastructure.	Information is saved and consistent among multiple clients and users.	All data/services are stored and run from a central Cloud database.	High	Achieved
5.4	Service supports multiple Users.	Multiple Users can be assigned to chatrooms.	XMPP MUC feature integrated..	High	Achieved
5.5	Chat messages will be archived for post-time access.	Messages are stored on server and can be accessed by contact name with Time-Stamp.	XMPP Flexible offline message retrieval integrated.	Low	Achieved

9. System Constraints

ID	Requirement	Success Criteria	Outcome	Priority	Status
1	Server that supports XMPP	Server supports the base functionality required for an XMPP server	Implemented OpenFire XMPP server.	High	Achieved
2	Client side implementation language will be Java	Pure Java Implementation is used.	All components Implemented in Java.	High	Achieved

10. External Interfaces

ID	Requirement	Success Criteria	Outcome	Priority	Status
1	OpenFire XMPP server	Server is running reliably, assuming host hardware is working	OpenFire Server Implemented.	High	Achieved

Changes For LCAM Resubmission

removed needs and features with priority level "out of current scope"

Usability Requirement 1.8 "user settings are restored" - removed

Performance Requirement 2.2 "The confidentiality of Messaging Data is ensured between the transit User devices." - priority changed to low

Performance Requirement 2.5 "Can be installed on popular OS." - changed to only specify desktop OS's

Administrative Requirement 1.8 "Accounts will be authorized as either regular user or administrator." - removed

User Requirement 2.1 "User data saved on client." - removed

User Requirement 2.3 "User devices can access Server from typical Internet enabled devices." - removed

User Requirement 2.5 "Usable for colour blind people." - removed

Security Requirement 4.1 "Service can support P2P Encryption." - priority changed to low

Changes For PRM Resubmission

Moved FR 2.1 Multiple Users can be added to Multi-User chats to 1.7 under usability.

Removed the follow FR because they were Repeated in NFR and not FR by definition:

- FR 2.2 The confidentiality of Messaging Data is ensured between the transit User devices.
- FR 2.4 The Service is reliable and has acceptable response times to Users.
- FR 2.5 Can be installed on popular Desktop OS.

Removed the following NFR repeated in FR and not NFR by definition:

- NFR 2.2 Usable for average people, without training necessary

Change NFR Quality Requirements to Maintainability requirements.

Changed NFR Administrative to Usability.

Moved NFR 2.1 User devices can run on Windows, OS X, and Linux environments. to 3.7

Moved NFR 3.2 Messages are exchanged between devices within 1 second to NFR 5.1.

Moved NFR 3.4 Users Client devices can be connected to Server through most traditional networks. to NFR 5.2.

Move NFR 1.4, 1.5, 1.6, 1.7 to NFR Security 4.5, 4.6, 4.7, 4.8.

Removed the following redundant NFR:

- 2.4 The Service is maintainable and extensible.
- 2.5 Clients can detect whether other users are online
- 3.5 High speed and quality development of client side application.
- 3.6 Risks avoided by using tried and tested components.

Move NFR 1.3 to 5.3, 1.1 to 5.4, 1.2 to 5.5.

Removed Constraint 4 The client side application must work on Windows, OS X, and Linux. Repeated in NFR.

Removed Constraint 3 The system shall be available 99.9% of the time. Not in scope of project.

Changes definition of NFR 3.7 to only support windows and possible support for OS X and Linux.

Changed Functional Requirements -> Usability Requirements to Chat Requirements.

Changed Functional Requirements -> Interface Requirements to Workflow Requirements.

Move FR 1.1, 1.2 to NFR 1.1, 1.2.

Removed FR 4.7 Submit drafted messages at will. Repeated in FR -> Chat.