Change History:

Date:	Version Update:	Member:	Description:
2013/08/22	1.0	fractals	Document created.
2013/08/24	1.1	fractals	Technical Requirements added
2013/08/24	1.2	fractals	Non-Functional Requirements added
2013/08/24	1.3	fractals	Technical Specification added
2013/08/24	1.4	fractals	System Features added
2013/08/24	1.5	fractals	Use Cases added
2013/08/24	1.6	fractals	Entity Relationship Diagrams added.
2013/08/24	1.7	fractals	Introduction added
2013/08/24	1.8	fractals	Purpose added
2013/08/24	1.9	fractals	Project Scope added
2013/08/24	2.0	fractals	References added
2013/08/24	2.1	fractals	System Description added
2013/08/24	2.2	fractals	External Interface Requirements added
2013/08/24	2.3	fractals	Open Issues added
2013/08/24	2.4	fractals	Glossary added
2013/08/24	2.5	fractals	Use Cases updated
2013/08/24	2.6	fractals	Functional Requirements updated
2013/08/24	2.7	fractals	Technical Requirements updated
2013/08/24	2.8	fractals	Open Issues updated
2013/08/24	2.9	fractals	Class Diagram added
2013/08/24	3.0	fractals	Glossary updated
2013/09/14	2.9	Janine Venter	Functional Features updated
2013/09/15	3.0	Janine Venter	Technical Specification updated
2013/09/15	3.1	Janine Venter	Project Scope updated
2013/09/15	3.2	Janine Venter	System Description updated
2013/09/15	3.3	Janine Venter	Support for Latex and Mimetex libraries updated
2013/09/15	3.4	Janine Venter	Messaging updated
2013/09/15	3.5	Janine Venter	Open Issues updated
2013/09/15	3.6	Janine Venter	Authorization updated
2013/09/15	3.7	Janine Venter	Scalability updated
2013/09/15	3.8	Michelle Peens	Entity Relationship Diagram updated
2013/09/15	3.9	Janine Venter	Use Cases updated
2013/09/15	4.0	Stephan Botha	Use case added for MimeTeX

Table of Contents:

1. Introduction

1.1 Purpose

This document is to serve as an agreement between our team of developers and our client Mr. Will van Heerden regarding the requirements for the Latex Chat application which will support users with their communication in a scientific and mathematical environment. This document will ensure that the requirements for this application is clear and agreed upon by all parties involved, and the development processes and phases to follow will be based on the requirements set out herein.

1.2 Document Conventions

- Document Formatting: LaTeX
- UML Diagrams: Diagram Designer, Visual Paradigm

1.3 Project Scope

The aim of the project is to develop an open source android XMPP chat client which supports the embedded LaTeX base equations which are rendered as images. LaTeX based equations will be rendered on the handset to produce mathematical equations. Our system will also provide the ability to view, edit and correct equations before sending. The application will provide a similar functionality to yaxim. Exchange of images and mathematical expressions will be possible through our software solution. The TeXchat application will have the ability to show a preview of the entered text send the equation as LaTeX code and then render it on the receiving end on the client handset.

1.4 References

- Mr. Will van Heerden.
- Android Authors, 2013. Android NDK.
  [Online] Available at: http://developer.android.com/tools/sdk/ndk/index.html
  [Accessed 18 August 2013].

2. System Description

The goal of our software application is to provide a chat service that will allow users to exchange messages and also to send mathematical equations in a rendered image format. The application is intended to provide a service to users that require the ability and support for a chat client that allows them to communicate more efficiently and effortlessly in a scientific, and mathematical context. It will provide a more usable mobile version of a Latex chat application.

Support for LaTeX and MimeTeX Libraries
The final application will have to make use of a Latex based library for the rendering of equations as images on the mobile device. For this reason we have implemented support for the MimeTeX library, through the use of the Android NDK (Native Development Kit), which allows us to embed the native C/C++ code of the MimeTeX library, in the source code of this release. The library is compiled through Eclipse using a special makefile that sets specific compiler flags to indicate that it is in math mode and not text mode.

Messaging
Our final goal is for messaging to be possible between multiple clients, and the support for sending Latex based equations, rendered on the client side and displayed as images on the handheld device. In this release however, our server supports two initial users, and have capabilities for them to send plain text messages and display these in a easy to understand manner. The ability to create, render, view and send mathematical equations have been implemented and is currently working. It is now possible to send and receive these equations in image format between the two clients. These messages should be stored statically on the device for later retrieval. For this purpose a SQLite database was implemented.

Login
A user should be authenticated by some type of login component. This has been implemented during the initial phases of development of this application. The server that our application uses provides the basic functionality for this authentication component, and uses a username and password based authentication method.

3.Functional Requirements

3.1 System Features

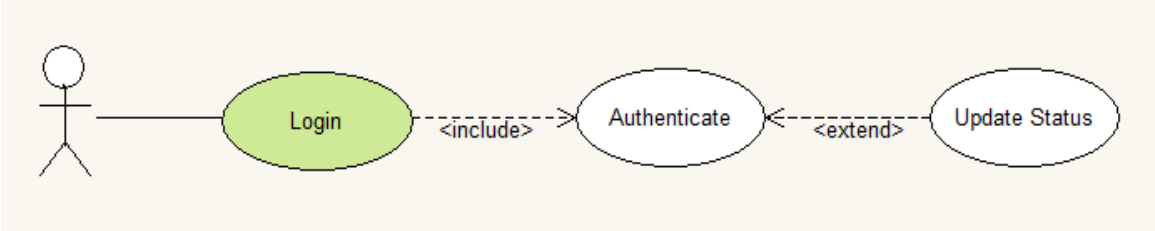
Login API
- Logging in to the application, authenticated by the server.
- Can remember your username and password if you activate the feature.
- Can cancel the login.

Viewing Contact List API
- Shows all of your contacts available on the server's contact roster.
- Display's each contact's status
- Allows you to interact with the contact list item and when you click on the contact name, it will open a messaging activity.

Message Sending API
- This is where the exchange of text messages happen between 2 contacts.
- Retrieves old messages from the database.
- Retrieves messages that were sent while the client was offline and displays them.
- Retrieves new messages sent by one contact to the other instantly.

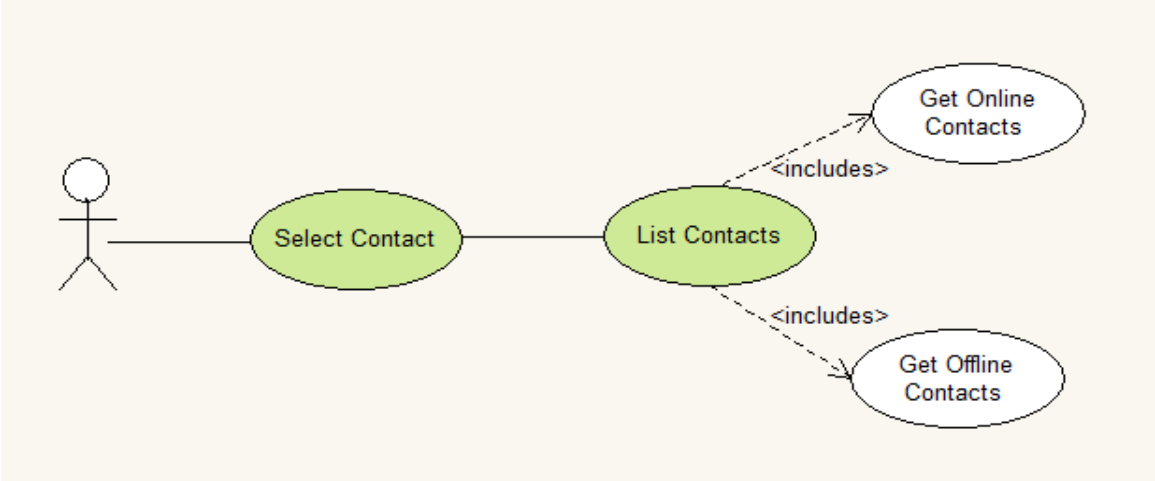
MimeTeX API
- Allows rendering of mathematical equations.
- Option to preview mathematical equation in image format.
- Renders image on both devices, after it was received and after it was sent.
- This API is used in the Message Sending API.

3.2 Use Cases Login



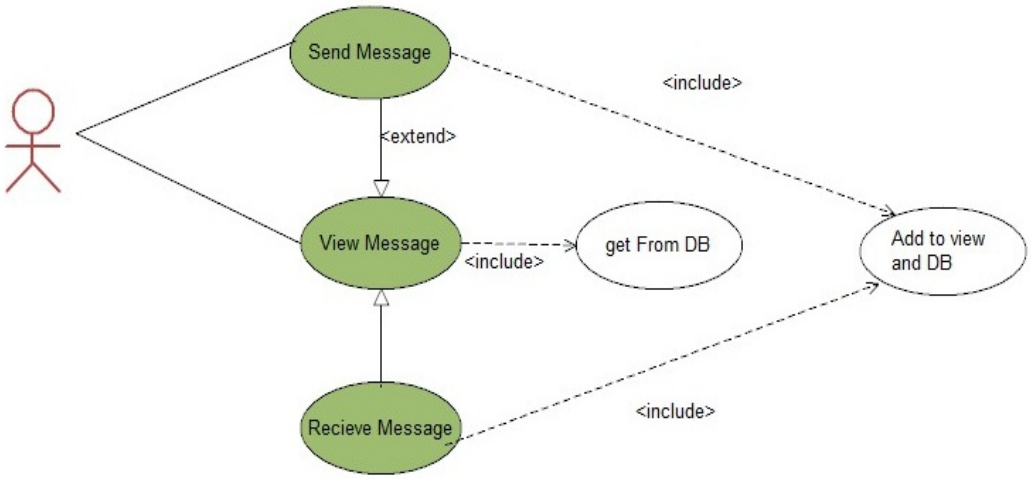
[Figure 1] Login Use Case

View Contact List



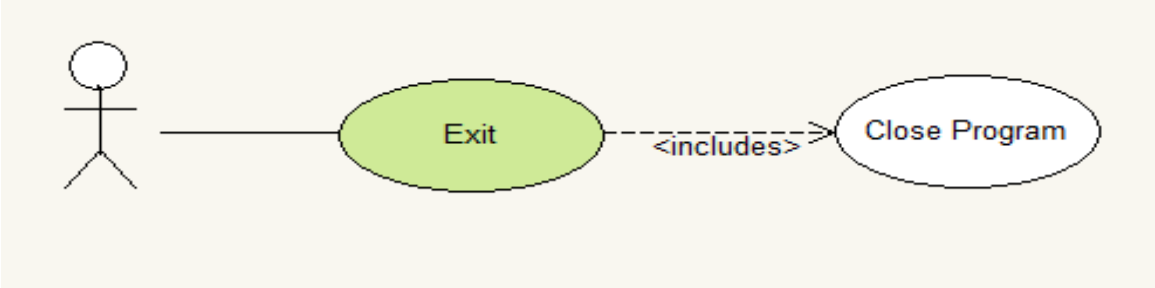
[Figure 2] View Contact List Use Case

Messaging



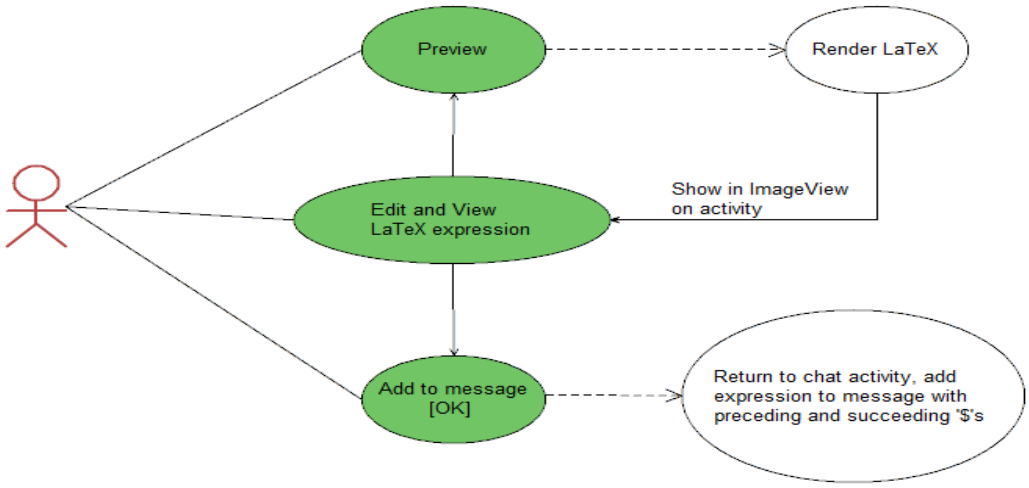
[Figure 3] Messaging Use Case

Exit



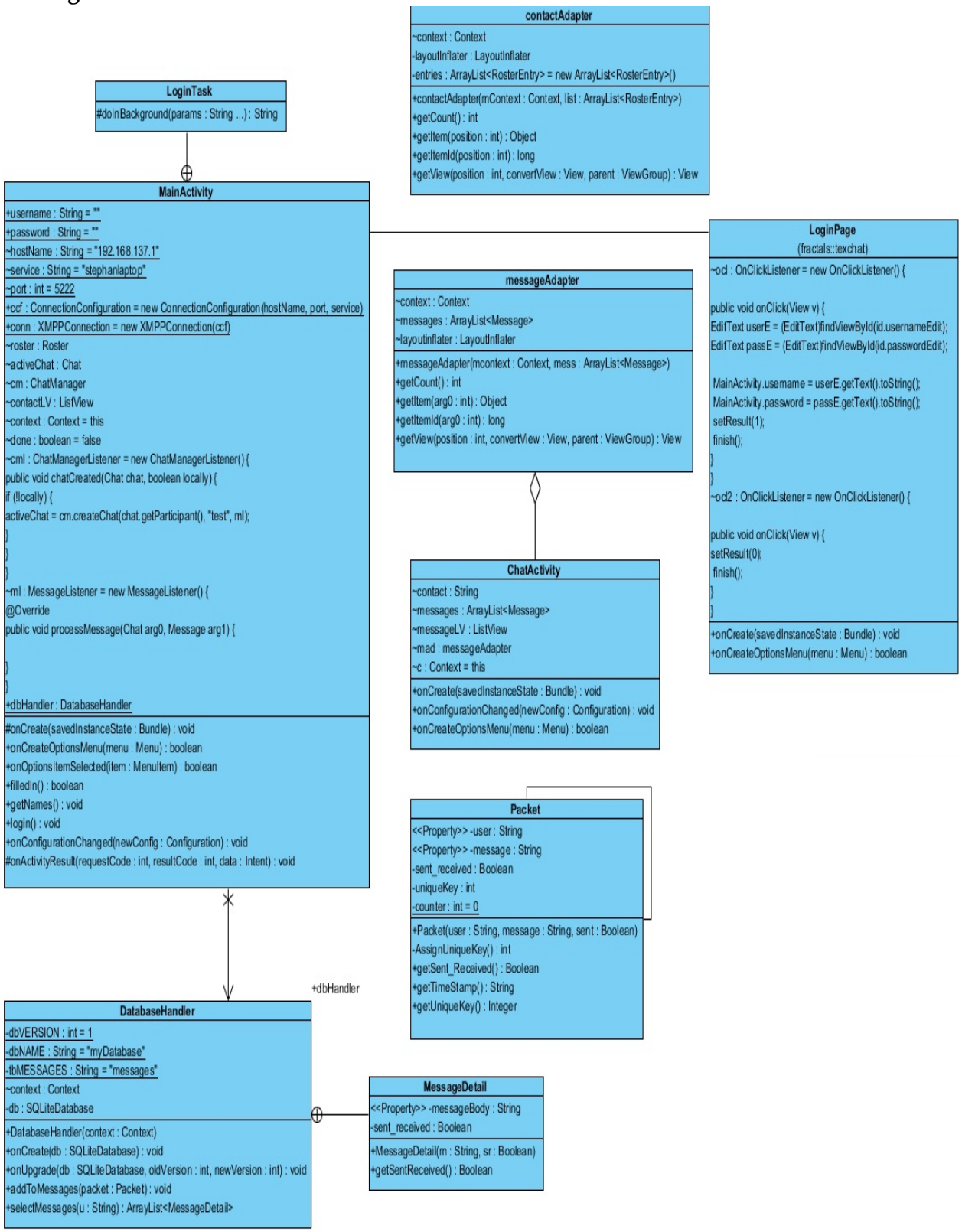
[Figure 4] Exit Use Case

MimeTeX



[Figure 4] MimeTeX Use Case

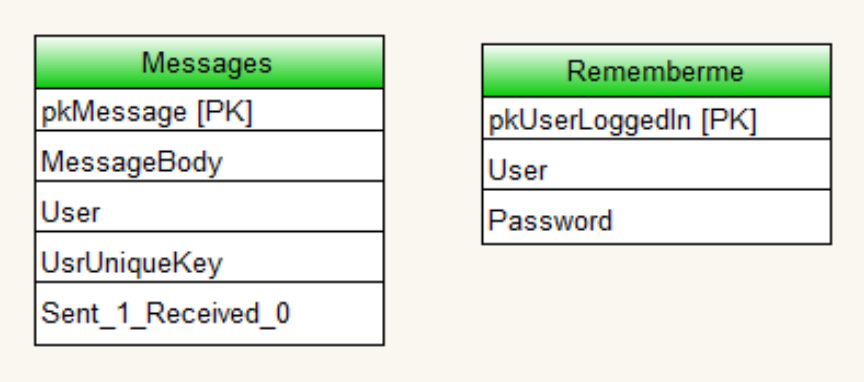
3.3 Class Diagram



[Figure 5] Class Diagram

3.4 Entity Relationship Diagram

The system makes use of one SQLite database on the client side. This database contains one table, namely Messages, that is used to record message histories for a particular client. This release does not include any database cleanup components, and these may expected from later phases of development.



[Figure 6] ERD

4. External Interface Requirements

- There is an external interface with the smack library.
- The mimeTeX library is also an external interface used within the application, the Native Development Kit acts as a bridge between Java and C/C++ when this library is used.
- The internal API uses the smack api to exchange the messages.
- The internal API is used to store data in the database on the device.

5. Technical Requirements(Non-Functional)

5.1 Non-Functional Requirements

Authentication
Authentication happens server side.
A login page is created to allow the user access to the application and its features.
As soon as the user's credentials has been authenticated he/she is logged into the application.

Authirozation
The authorization is in progress and will be completed by the next iteration.
Scalability
At this point in time the system is developed to handle only 2 users at a time. It will later be scaled upwards to handle a larger amount of concurrent users. A solution to the scalability is by using an online server that will handle more than two clients concurrently

Audit-ability
The messages are logged on the phone's database created by the application as soon as it is installed.

5.2 Technical Specification
Platforms to be supported:
- Android

Will use both Java and C/C++. C/C++ will be used as native code. Is developed to communicate with a local server The database is handled with SQLite.

6. Open Issues

- Online server functionality
- Visual elements and styling

7. Glossary

- SDK - Software Development Kit
- NDK - Native Development Kit
- ADT - Android Development Toolkit plugin
- IDE - Integrated Development Environment
- RUP - Rational Unified Process
- XML - eXtensible Markup Language
- Agile - Development methodology
- MVC - Model View Controller
- UML - Unified Modelling Language
- API - Application Programming Interface
- MimeTex - LaTeX library ported to Android
- C - Programming language.