UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

Department of Computer Science
Faculty of Engineering, Built Environment & IT
University of Pretoria

COS301 - Software Engineering

Atbash

Software Requirements and Design Specifications

June 2, 2021

Item: Capstone Project - Demo 1

Team Name: Bit by Bit

Team Members:

Name	Surname	Student Number
Liam	Mayston	u19027801
Dylan	Pfab	u19003961 *
Connor	Mayston	u1906936
Joshua	Reddy	u19196042
Targo	Dove	u15020275

** - indicates team leader*

Contents

1 Introduction

1.1 Project Owner

Mr Reinhardt Eiselen - email
Mr Matthew Gouws - email
Mr Peter Rayner - email
Amazon Web Services - email

1.2 Purpose

The purpose of this document is to present an overview the proposed Atbash system. This document will describe the target audience, user-interface, functionality and requirements (hardware, software and business). This document is intended for EPI-USE, the stakeholders, and the developers who will implement the system.

1.3 Vision

Our vision is to create a secure, fast and easy to use messaging application that will enable secure communication for all users.
Atbash is a messaging application, where the privacy of messages is the top priority. Message content is only visible to the sender and the recipient and can never be seen, even by the owners.

1.4 Objectives

- Provide a Mobile app for sending/receiving messages
- Provide peer to peer encryption, where only persons communicating can see messages
- Ensure the system is able to scale automatically as demand grows

2 User Characteristics

The user should be operating a smartphone to download and use the application. The user should have access to the internet and understand how to message. The Atbash system will be used by the following users:

2.1 Regular User

- Does not require any technical knowledge of the system or other similar systems in general.
- Is familiar with their smartphone and with messaging as a concept
- Is not familiar with the concept of end-to-end encryption

3 Functional Requirements

The requirements of the system models the functionality that the Atbash system offers and should allow for the following functionality:

R1 : The system should allow the user to register an account and link their phone number

R2 : The system should allow the user to login securely

R3 : The system should allow the user to change their display name, status (online/offline/busy/away/custom), and profile picture

R4 : The system should allow the user to add contacts

R5 : The system should allow the user to send private messages to their contacts

R6 : The system should send push notifications if the application is not open

4 System Domain Model

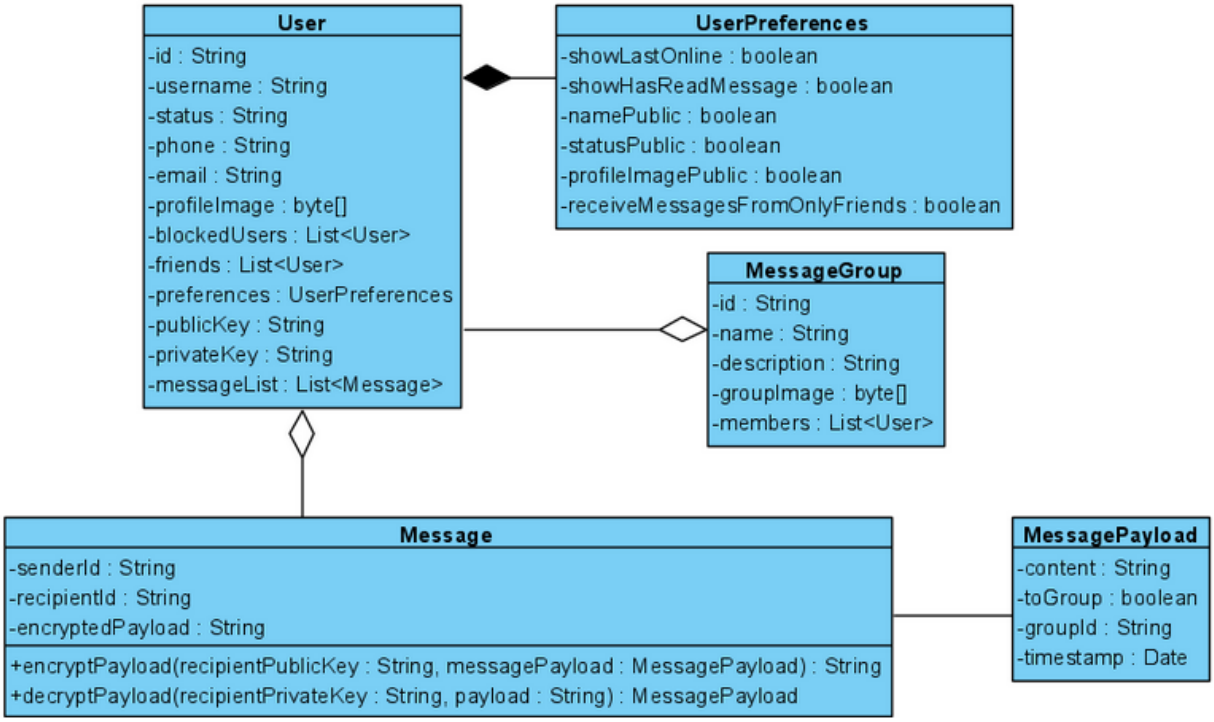

Figure 1: Diagram showing the System Domain Model for the Atbash system

4.1 Description

The Atbash system consists of only a few classes. The User class is databse of all of

5 Quality Requirements

Q1 : Security - The system must be secure. A user should not be able view another user's information. No one except the sender and recipient should be able to read their conversation.

Q2 : Scalability - The system should be scalable. An increase in the number of active users should not significantly slow down the performance of any particular user.

Q3 : Availability - The system should be able to be accessed anywhere anytime. This will require the system to have a down time of less than 1 hour in the case of an emergency or fault.

Q4 : Usability - The system must be usable. The regular user is assumed to be familiar with smartphones, but not with the concept of encryption. The system must educate the user about the encryption that the system uses.

Q5 : Flexibility - The system will be accessible for technological upgrades and updates in order to stay relevant with future technologies.

6 Subsystems

Below follows a description of the various subsystems that the system is composed of. Together with this the scope of each subsystem is represented by a use case diagram. For each use case a service contract can be designed which details preconditions, post conditions, in-variants and interactions that each use case has in the system.

6.1 User

The user subsystem involves all the actions a user may take when operating Atbash, including login and registration, updating account settings and starting new chats or adding contacts.

6.1.1 Use Cases

The use cases of the user subsystem are shown in Figure 2, showing all the functionality a user should be able to do in the system.

U1 **User Subsystem Use Cases**

U1.1 **Login:** The user will be able to login using their account credentials, the data will be checked securely to see if the user does exist and then log them in.

U1.2 **Register:** The user will be able to register a new phone number and password to create an account. The system will validate that the user does not already exist before creating the account.

U1.3 **Update Settings:** The user will be able to update their profile picture, status and display name in a settings screen.

U1.4 **Start New Chat:** The user will be able to start a new chat with one of their existing contacts.

U1.5 **Add Contact:** The user will be able to add a new contact to their list.

U1.6 **Send Message:** The user will be able to type a message and send it to someone in their contact list.

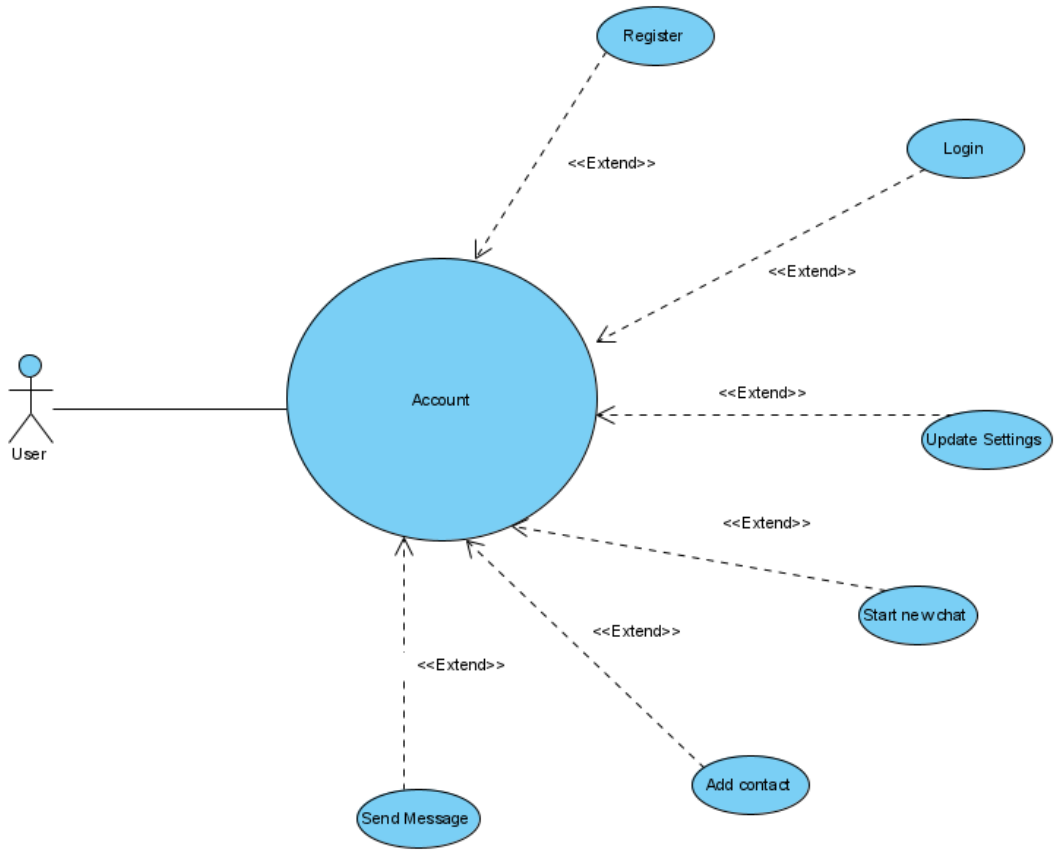

Figure 2: Diagram showing the use cases for the user subsystem

6.1.2 Domain Model

The user domain model is quite brief but is vital as a lot of other subsystems will make use of it.

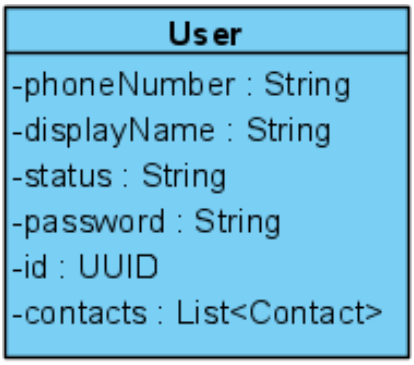

Figure 3: User Domain Model

6.1.3 Service Contracts

To make the system modular and more maintainable service contracts are created to allow for pluggability and easy replacement.