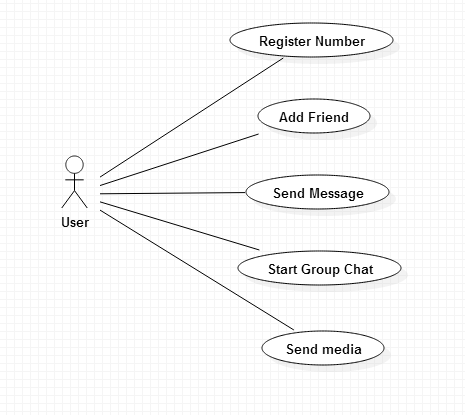
**Technical Architecture**

Gary Morgan

X00093748

1 Use Cases



|  |  |
| --- | --- |
| Title (goal) | Register Number |
| Primary Actor | User |
| Scope | User has network connection  User has a phone number |
| Level | High Complexity |
| (Story) | The user will input their phone number which will then have a confirmation code sent to it via over the mobile network.  The user will have to type in the confirmation code to be registered and they will be logged in from then on. |

|  |  |
| --- | --- |
| Title (goal) | Add Friend |
| Primary Actor | User |
| Scope | User has network connection  User is logged in  Friend also has account |
| Level | Medium Complexity |
| (Story) | User will input their friends phone number who will then receive a proposal to accept the friend request. This will add them to each other’s list and allow them to begin a conversation with each other. The contacts will be saved locally. |

|  |  |
| --- | --- |
| Title (goal) | Send Message |
| Primary Actor | User |
| Scope | User has network connection  Both users are contacts |
| Level | High Complexity |
| (Story) | The user will select which contact they wish to message. This will either open their conversation window or make a new one.  User types in message and sends. Message is encrypted using AES encryption and given a key by the server while being sent via https. Other user receives message and decrypts with the same key. |

|  |  |
| --- | --- |
| Title (goal) | Start Group Chat |
| Primary Actor | User |
| Scope | Users have network connection  User who starts chat has all numbers |
| Level | High Complexity |
| (Story) | One user creates the room and chooses who to add to the room from their contact list. All users are instantly added rather than being invited in case the creator needs to send an important circular message at that time. They have the option to close the conversation at any time. |

|  |  |
| --- | --- |
| Title (goal) | Send media |
| Primary Actor | User |
| Scope | Users have network connection  Users have each other as contacts  User has media stored locally |
| Level | High Complexity |
| (Story) | User chooses the option to send media (video or picture) and then chooses from their local files which media file they want to send. Files is encrypted and decrypted in the same fashion as a normal message above. |

2 Technical Architecture

2.1 Software Components

The app will be written using C# on Microsoft Visual Studio using the windows phone emulator that is included in Visual Studio. Databases will not be needed as the conversations are stored locally and user accounts are unique to the phone number of each user. A server will be used for encryption and key generation between users sending messages. The app will initially be developed for the Windows Phone and distributed using the Windows Phone Store. Messages can be stored on a server up until delivery for reliability. The app will not use end-to-end messaging and will instead use https messaging, with the message going from peer-to-server-to-peer.

2.2 Platform libraries

The app could use an ASP.NET web API which could be secured with Hash-based message authentication code within HTTPS. I chose this method as it is the same authentication method that Amazon Web Services use and many other libraries implement it. It creates a password from random data in HTTPS requests like the timestamp and HTTP verbs. This will all be used with C# as the language.

2.3 Distribution and Deployment

The app will not need to use any cloud storage as all conversations are stored locally on mobile devices but a https server will be used to provide encryption services which will make it easier for key distribution. The app will be distributed using a RESTful web service as it is faster than SOAP and is the distribution service that most new applications are using. It doesn’t require middleware or XML messages to operate.

For security I will be using HTTPS on the server-side to handle key distribution and encryption. HTTP cannot be used on an app that should be secure as it is prone in particular to man-in-the-middle and eavesdropping attacks which HTTPs is considered secure against. HMAC will be used in HTTPS for authentication in the app.

2.4 Risks

Unfortunately at this moment the app will be using a server and it will not be 100% guaranteed privacy as the server operator will be able to access messages as it generates the key itself. The main risks with this app is the chance of security breaches or a potential intruder being able to eavesdrop on the message before it reaches the server for encryption or if in some way someone was able to retrieve the key to a conversation. This could be lessened by a key that is reset at timed intervals so if someone manages to get a key it will not last long. However it is more important that this doesn’t happen in the first place which is why I must implement the security that I mentioned above.

13 Prototype

3.1 Prototype Deliverable for week 8

For my prototype for week 8 I hope to at least have the use cases Register Number and Add Friend done. I could possibly have a part of message encryption or key generating as well. Register number can be tested by inputting a phone number that belongs to me and checking that I receive the code properly. I can then login to see if I have been successfully registered.

3.2 Prototype Deliverable for week 11

For week 11 I hope to have message sending done along with encryption and decryption working fully. I feel this is the main functionality required and also the most time consuming so if I have message sending fully operational adding in the extra features should be relatively quick.