

Link

Get connected

Link App Design Document

Justin Richard

Version 1.1

September 20th, 2016

Revision History

Date	Version	Description	Author
Sept 20th, 2016	1.0	Initial version of document	Justin Richard
Oct 15th, 2016	1.1	Review after development started	Justin Richard

Table of contents

Introduction	4
Purpose	4
Scope	4
Definitions, Acronyms, and Abbreviations	4
Use Cases	5
Actors	5
Link App	5
App user	5
Case 1 - Pair of users	5
Brief Description	5
Goal	5
Success Measurement	6
Precondition	6
Event flow	6
Case 2 - Group of users	6
Brief Description	6
Goal	6
Success Measurement	6
Precondition	6
Event flow	6
Case 3 - User alongside new user	7
Brief Description	7
Goal	7
Success Measurement	7
Precondition	7
Event flow	7
Design Overview	7
Introduction	7
System Architecture	8
Database Structure	8
System Interfaces	8
User interfaces	8
Software interfaces	10
Constraints	10
Requirements	10
Functional requirements	11
Non-functional Requirements	12
	3

Introduction

This software design document is intended to provide information which will be used to aid in the development process of the Link application. The idea for this application came from a family vacation away to Toronto. While away, multiple times we were caught in extremely busy locations that were desirable to split apart in. The first was a mall, and second was a smaller amusement park. The problem with splitting up in a place like this is in order to get back together you either need to wander aimlessly, or agree on a time and place to meet up at which can waste time and cause issues if plans change. This problem can even be as simple as trying to find a friend on University campus. A few times a semester while working in the computing facilities I run into people who are unable to find their friend. Meet Link: Link is application designed to help you efficiently find or meet up with your friends or family. Link allows you to share real time updates on current locations to effectively connect with others. You choose who and when you want to Link with. Applicable scenarios will be ran over in the use cases section.

Purpose

The purpose of this document is to provide a description of the system to a full enough extent such that a third party could understand and develop this software themselves. This document should include narrative and graphical forms of documentation, as well as models of the system. The application itself is being completed as part of UVic's SENG 490 - Directed Studies course, as well as completed off of personal interest.

Scope

This design document is designed to give an outline of the system and how it will work, but not to declare down to the button how it will be. It should provide a solid baseline which can be later built upon as requirements and goals change through development and testing.

Definitions, Acronyms, and Abbreviations

App	Short for "application". A program installed onto a device.
4G	A mobile communications network, used to connect a mobile device to the internet.
OS	Short for "Operating system". In this case would be either Android or iOS.
AWS	Amazon Web Services.

Use Cases

Actors

There are only a small amount of unique actors in the use cases for Link. The separate users of the application need not fit into different “roles” of users, and therefore can be grouped into one generic “User” actor. The other acting entity will be the application itself. The app will take actions based off of user actions which makes it a valid actor.

Link App

The app actor is the Link application. It does actions in correspondence to actions completed by the users and scheduled tasks.

App user

Any user of the app will fit into the user actor role. Users of all intents will still interact with the application in the same manner, therefore all fit the same actor. The reasons behind their actions and use of the app may be different, but all end results are the same.

Case 1 - Pair of users

Brief Description

This use case describes almost all major interactions with the application. A set of 2 users will connect with each other, and complete a Link session to locate each other.

Goal

The two users are able to find each other's locations with assistance of Link.

Success Measurement

The users were able to locate each other easier/quicker than if they had not used the app, and the users require less communication with each other to accomplish the task.

Precondition

- Both users have the app installed
- Both users are already contacts with each other

Event flow

1. User A and user B want to meet up, so user A opens the Link app finds B in their contacts and selects ‘Link up!’.

2. User B receives a push notification that user A wants to Link up with them. User B accepts this request.
3. User A and B can now see real time updates on each others GPS location overlayed onto a Google Maps interface.
4. Users A and B successfully find each other.

Case 2 - Group of users

Brief Description

A collection of users (Sports team, friend circle, work group etc.) wants to create a group Link session that can be joined at any time with any amount of people.

Goal

The users in the group are able to find each other's locations whenever the users wish to.

Success Measurement

The group sessions can be considered a success if the users prefer using group sessions over direct Link sessions for finding people in their group.

Precondition

- All users have the app installed
- All users are either contacts, or have a direct connection through a contact

Event flow

1. User A wants to create a group session for their group, so they open the menu and select 'New group Link'.
2. User A selects an initial pool of contacts to invite to the group.
3. Users B, C, D accept the group invite.
4. User C invites his contact user E to the group which they accept.
5. Users A, B, C, D, E can join the Link session at any point in time which will broadcast their location, and allow them to see the other users locations when >1 user is active.

Case 3 - User alongside new user

Brief Description

This use case describes a Link app User A that wants to do a Link session with someone who does not have the Link app.

Goal

The users are able to get into a Link app session with minimal overhead required to get the second user setup and going.

Success Measurement

Success can be measured by determining the amount of steps required for the user to add their friend as a contact, and how large the app file size is so that the new user can download it quickly and even over cellular data.

Precondition

- One user already has the app
- The other user has a smartphone and an internet connection

Event flow

1. User A and the new user, user B want to start a Link session
2. User A can use the share option to send the app store link to user B, or user B can manually search in the app store for the Link app.
3. User B downloads the app and signs in with one of the social media providers.
4. User A and user B can add each other as contacts through the add a contact menu which lists a user's friends from Facebook or Google accounts that also use the app.
5. The users can now continue with use case 1.

Design Overview

Introduction

The design overview section is designed to give you a preview of how the application will be laid out, and how it might look. It will also give you an understanding of how the back end architecture will work to make the app function.

System Architecture

The backend of the system is going to rely on a spread of AWS services. Initially several application servers were going to be made to handle various tasks as can be seen in figure 1.0, but after some thought it was determined that AWS was a perfect fit for the job, and would be much cheaper. API Gateway will be used to expose an API to the public which the device can interact with. Cognito is used for identity management to connect users and their social media accounts to a single identity to use for this app. An authenticated Cognito user can use the STS service to get an access token allowing calls to the API Gateway. API calls to the API Gateway will trigger different Lambda functions based on the intent, and the Lambda functions may update/read/delete items in the DynamoDB database. You can see a general outline of the connections in figure 1.1.

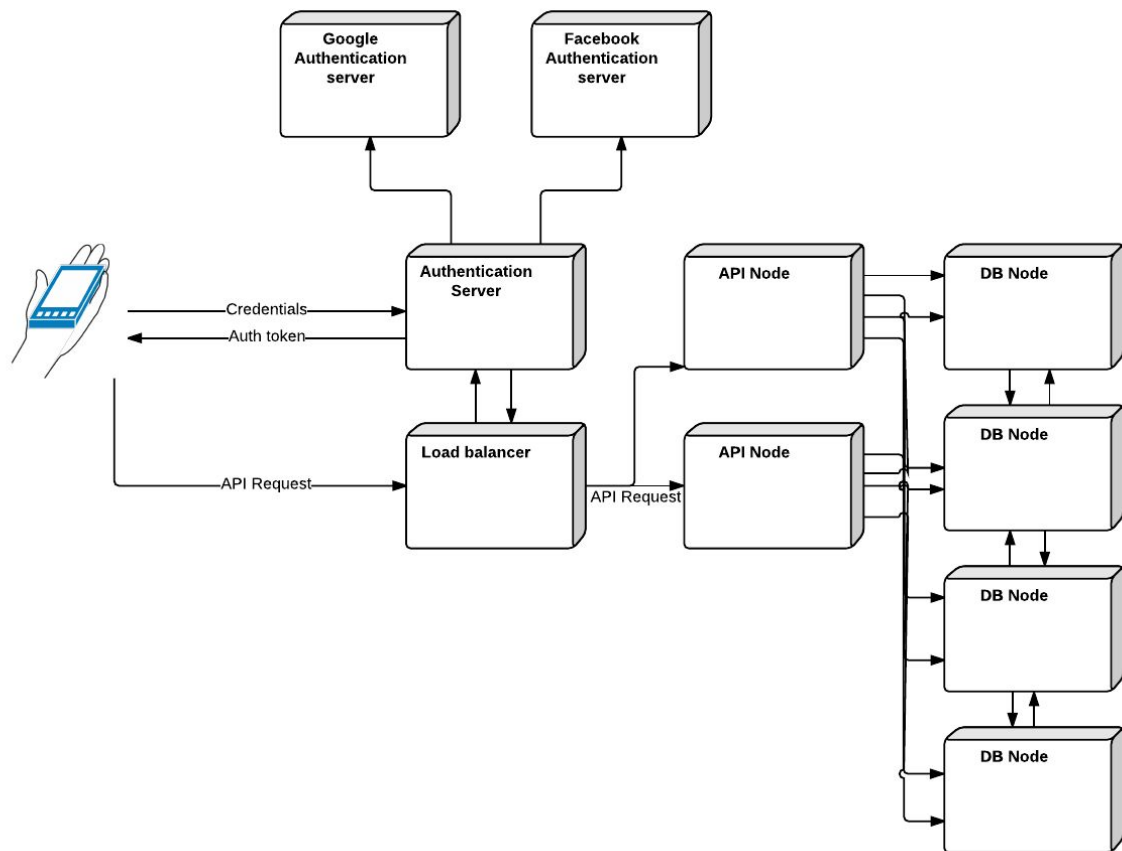


Figure 1.0 Initial System Architecture Concept

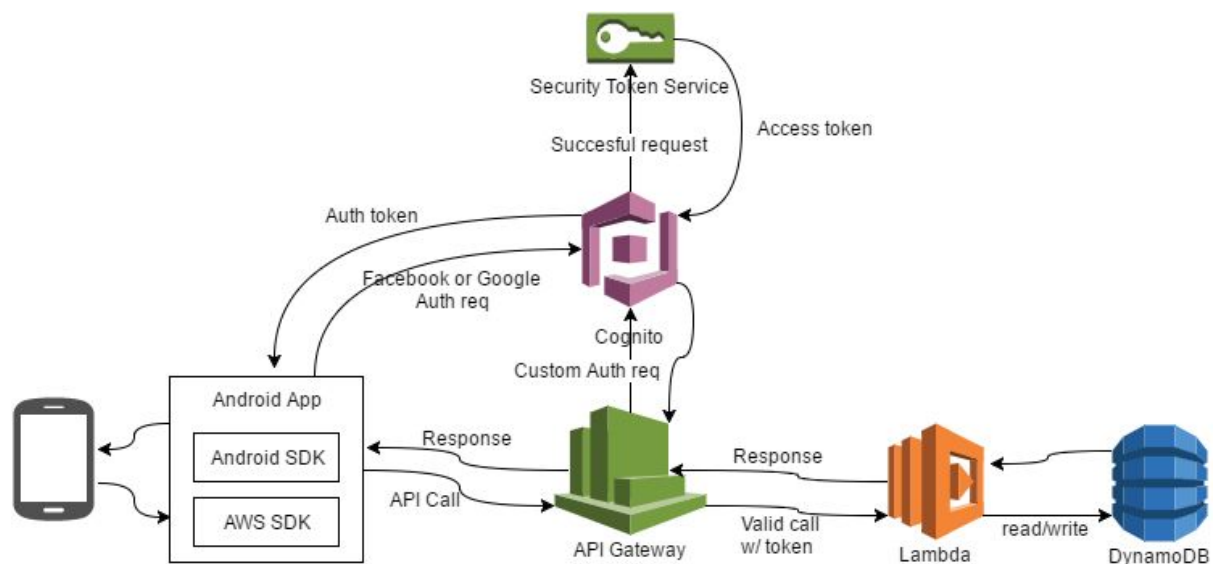


Figure 1.1 High level System Architecture

Database Structure

The expected database structure is fairly small and straightforward. There will be many users, each of whom will have many contacts which belong to them, and many participant objects. Participant objects each belong to one Link object. The various properties of each of

these objects can be seen in figure 2. The Link object holds meta information about the Link session, the participants objects hold the data about each user's location in the Link session, and the contact objects hold the connections between users.

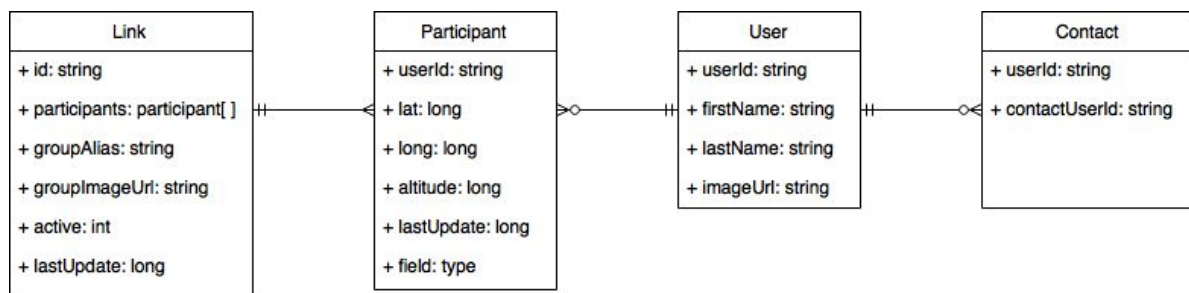


Figure 2. Database table structure

System Interfaces

User interfaces

There are three main user interfaces in the Link app, as well as a few small subinterfaces. The main home screen will consist of a layout very similar to that of Facebook Messengers chat application. Users are likely to be familiar with this layout, and it has proved to be efficient with high reviews. The main key interfaces are prototyped below in figures 3 to 6. Selecting one of the Links from the Links tab will bring you to the interface in figure 6, showing an active Link session.

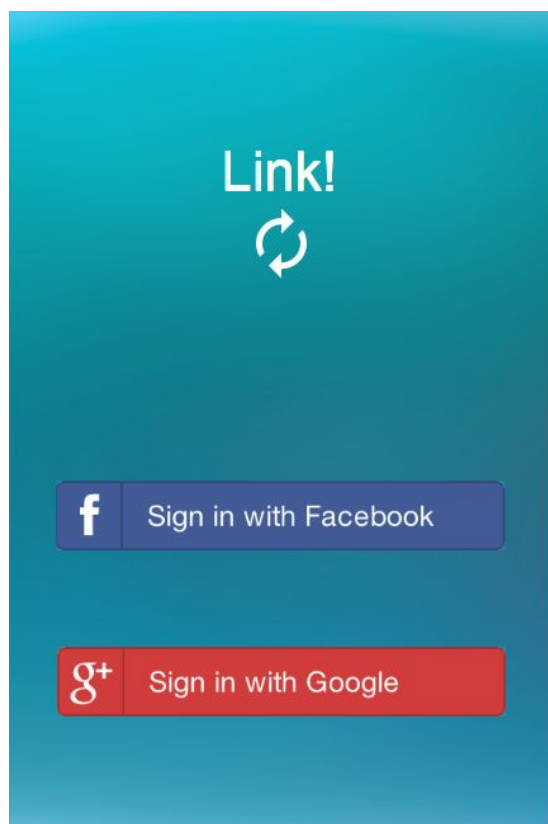


Figure 3. Sign in page

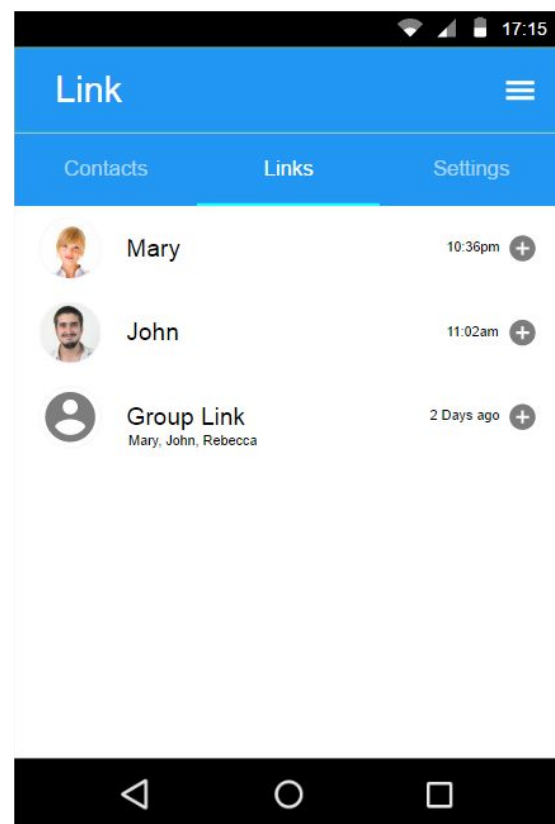


Figure 4. Home screen

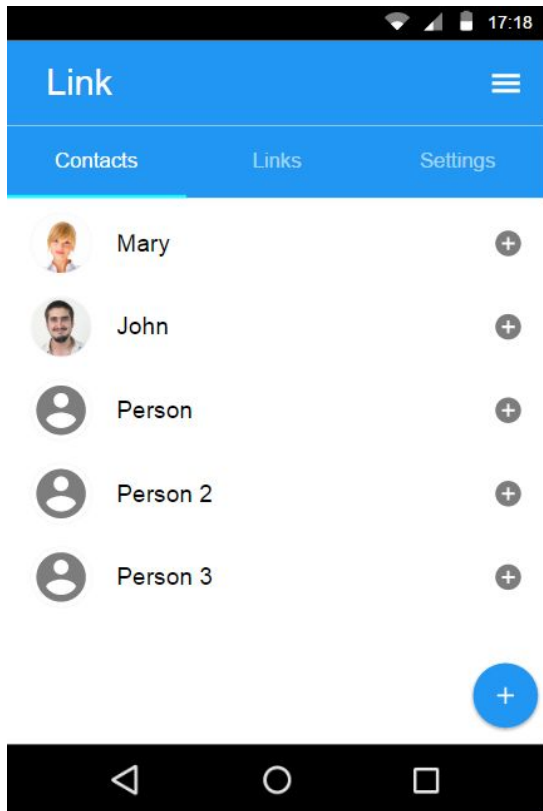


Figure 5. Contacts tab

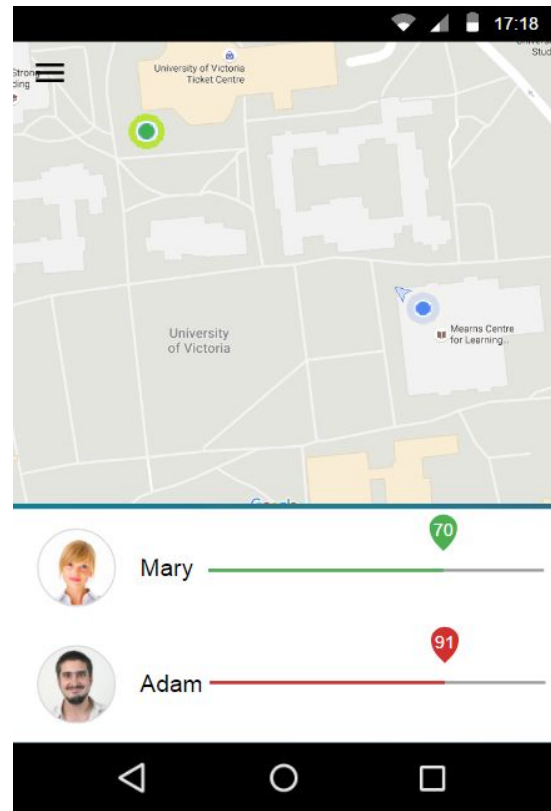


Figure 6. An active link session

Software interfaces

The application will have a software backend interface that is not exposed to the client. This will take the actions and interactions from the user and form the data into appropriate forms for the database. It will connect with AWS infrastructure to complete transactions of data. The main component of this will just be to update a user's contacts and data about active Link sessions in the database.

Constraints

- Google Maps API functionality may limit desired map data
- Android SDK for Android version of the app may limit certain features
- iOS SDK for iOS version of the app may limit certain features or development tools (Mac computer is required)
- Time: The first version of the app is scheduled to be completed by mid December at the latest in order to fit academic course
- Money: iOS development and release requires a fee of 100\$ combined with the cost of obtaining a Mac computer, therefore the first version of the app will be for android only.

Requirements

The Link App has a large set of requirements designed to make the app desirable to use by the largest amount of users possible. The functional requirements, what the app should do, and the non-functional requirements that define how the app should do it can be found in the following two sections. These sections may be altered as development of the application proceeds. The priority of the requirement is also listed. The priority can be either low, medium, high, or essential. Low and medium priorities are reserved for requirements that would be nice to have, but are not key parts of the app.

Functional requirements

The functional requirements break down what the app should be able to do, and can be found in table 1.

No	Identifier	Description	Priority
1.0	Authentication	Users must be able to log into the app with a major social media provider	Essential
2.0	Contacts	Users should be able to add other users as a contact	Essential
2.1	Contacts	Users should be able to remove contacts	High
2.2	Contacts	Users contacts should be saved online and locally to save network requests	High
2.3	Contacts	Users devices should request a delta of the changes to their contact lists to minimize data usage	High
3.0	Sessions	Users should be able to create a Link session with any of their contacts	Essential
3.1	Sessions	Users should be able to create a group Link session with any of their contacts	Essential
3.2	Sessions	Users should be able to be invited to a Link or group Link session by a contact	Essential
3.3	Sessions	Users should be able to see real time updates of the other users locations in an active Link session	Essential
3.4	Sessions	Users should be able to leave an active session to make it inactive	Essential
3.5	Sessions	Users should be able to leave a group session	High
3.6	Sessions	Users devices should request a delta of the changes to	High

		the list of their sessions to minimize data usage	
4.0	Customization	Users should be able to select between different themes	Low
4.1	Customization	The app should be tailed to the user with welcome messages and show their name and picture	Medium
4.2	Customization	Users should be able to set preferences about their data usage and notifications	High

Table 1. Functional requirements

Non-functional Requirements

No	Identifier	Description	Priority
1.0	Authentication	Users should be able to login with Facebook or Google	Essential
2.0	Contacts	Users should be able to view a list of Facebook or Google contacts who use the app, and be able to add them as Link contacts	High
2.1	Contacts	Users should be able to add a contact directly by a unique identifier displayed in the contact menu	Medium
2.2	Contacts	Users should be able to invite people to be their contact by sharing via the OS's native sharing features	High
3.1	Sessions	In active Link sessions users should be able to see current distance to each participant, and a visual overlay in Google Maps	Essential
4.0	Customization	There should be two themes available, a default light one and a selectable dark theme	Low
5.0	General	Network requests should take no longer than 250ms to complete on a stable 4G or WiFi connection	High

Table 2. Non-functional requirements