



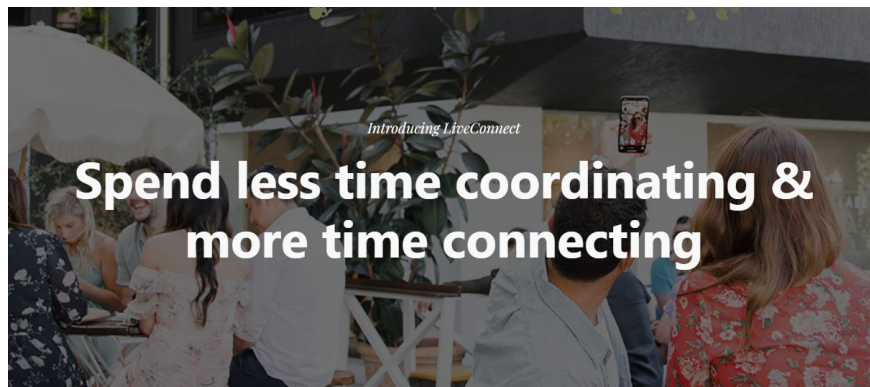
# LiveConnect Feasibility Report

Corey Rice  
Jacob Stone  
Jon Hoeve  
Jhoseph Ruiz



# Overview

- Cloud - based iOS application
- Organize social gatherings easily
- Messaging
  - Person to person
  - Groups
- Create polls
  - Venue
  - Time
  - Other details
- Web interface



<https://liveconnect.ca/product/>

# Languages

- Swift - main language for iOS application
- JavaScript - web based aspects
- React - web based aspects
- HTML - structure and layout
- CSS - structure and layout
- Kotlin - main language for Android application



# Libraries, API, SDK

## Core Location

We are going to use Core Location to determine where a device is so we can have establishments nearby pop up.

## Cocoapods

Cocoa pods will be used to help the app to work on multiple devices without having too many dependencies.

## Firebase

We will be using Firebase to collect and store data for the use of providing a more catered experience.



Firebase

## MapKit

We will be using MapKit to help point out places of interest nearby the user. This will be helpful when it comes to suggesting establishments or meeting places.

## MessageKit

MessageKit will be used with the messaging between people whether it's in a group or person to person.

## iOS chat SDK

iOS chat will be used to automate the messaging process as well as collect data from chats.



# Code Repository Organizations

## Branching

- 6 different branches, Main, Development, Documentation, Feature, Enhancement, and Bug.
- All work together under one github repository to optimize our work towards the final prototype.

## Main

- Main is the branch that contains the most up-to-date working prototype
- Pull requests to this branch must be reviewed by more than one group member.

## Development

- Bug free code that is still being worked on by group members.

## Documentation

- Any documentation that relates to the project.

## Feature

- Any code that adds big functionality to the application.

## Enhancement

- Improving on code that does not have any known bugs or issues.

## Bug

- Code that needs to be worked on because has plenty of bugs.



# Code Repository Organizations

## Zenhub

- Allows the whole group to visualize what is being worked on and what still needs to be done.

## Icebox

- Work that is not being done in this sprint is placed.

## Backlog

- Work that needs to be done in this sprint but has yet to be worked on goes.

## In Progress

- The issues that are being worked on.

## Review Q/A

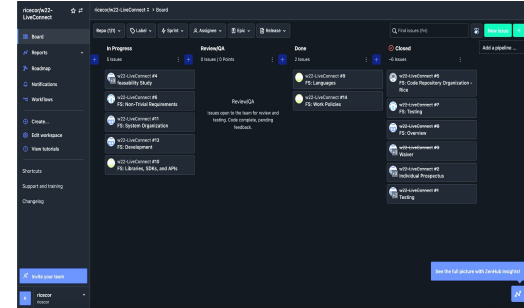
- Work that has been completed and is in need of review from the group.

## Done

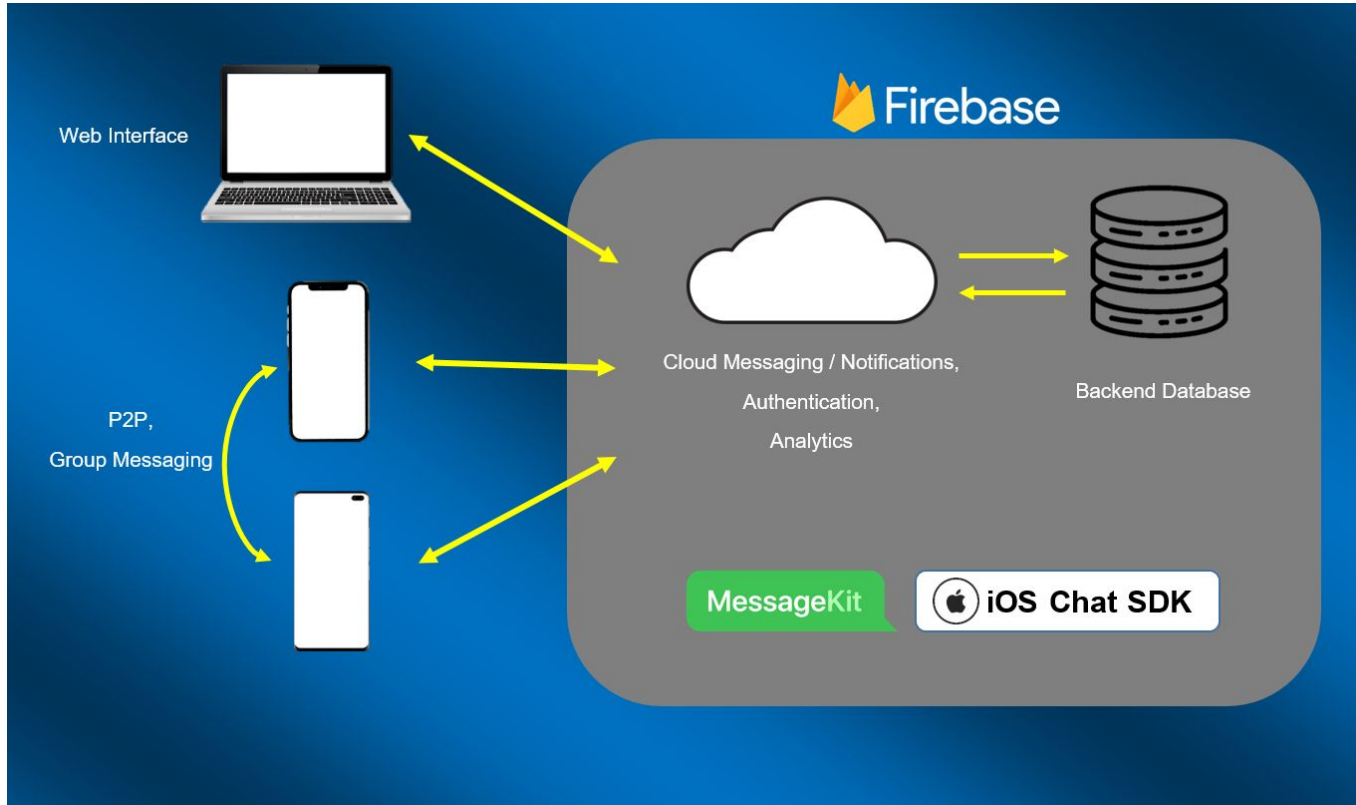
- Work that has gone through the review process.

## Closed

- Issues that have been pushed to the main branch goes.



# System Organization



# Development

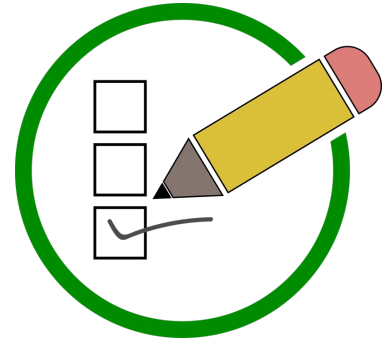
- Application: Xcode for Development
- Github: Version Tracking
- Zenhub: Agile Project Management
- Clockify: Work Progress Tracking





# Non-Trivial Requirements (User)

- Messaging System: iOS Chat SDK, Firebase, MessageKit
  - P2P and Group Message Capability
- Polling Feature: Firebase, Analytics
- Booking Feature: Firebase

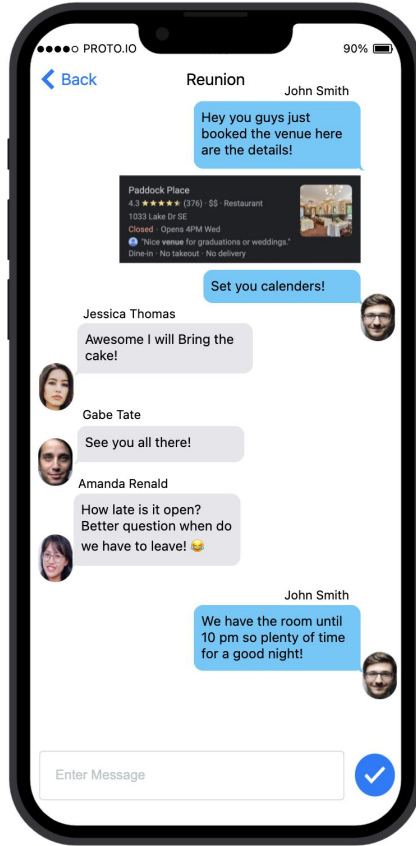


# Non-Trivial Requirements (Establishment)

- Web Application: HTML, JavaScript, CSS
  - JavaScript for Coding and HTML and CSS for structure of website.
- Bookings List (JavaScript)
  - View bookings for establishment given from users
- Management Features (JavaScript)
  - Update Availability and Bookings to limit bookings if needed



# Mock-Ups



- Users will be able to chat and share locations for the event through Google.
- Users will have names and a picture that will be displayed for the group to see as shown.

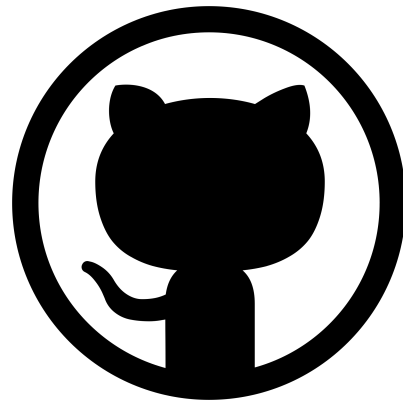
# Work Policies

- Corey: Lead Programmer
  - In charge of most of the programming and assigning tasks
- Jhoseph: Messaging aspect of the application
- Jon and Jacob: Varying roles depending on the need.
  - All documentation and presentation.
- Testing: done as a group
- Flexible work policies allowing for quick changes when needed



# Testing

- Individual Testing
  - Each member develops their code to pass all unit testing
  - TestSuite in Xcode
  - LiveServer for Web Application for Visual Testing
- Pull Requests
  - Team must review new code ready for merge
  - At least one team member must approve before merge



# Deliverance and Milestones

- Sprint 1
  - Login Screen
    - Authentication
    - Invitation Feature for new users
  - Messaging Service
    - P2P messaging
  - Database
    - Store and change data
  - Template Design
    - New events template
- Sprint 2
  - Database Analytics
  - App Features
    - Search Feature
    - Polling Feature
    - Booking Feature
  - Notification Services
  - Events List
- Sprint 3
  - Messaging Service
    - Group based messaging
  - Invitation Feature
    - Invite through diff. medias
  - Web
    - Prototype website page
  - Search Feature
    - Filter options for searches
- Sprint 4
  - Web Application
    - Polish features for better UX
  - Polling Feature
    - More options for polling
  - Events List
    - Filter & Search features
- Sprint 5
  - Documentation
  - Testing
  - Bugs
- Sprint 6
  - Presentation
  - Product Submission

*Thank You*



**LIVE CONNECT**