

# Kite : A Social Media Journal App

Project Sponsored By:

David Vasquez

## Requirements Document

Written by:

David Baugh

Sawyer Kokesh

Andrew Bowers

### **Abstract:**

This Requirements Document provides an in-depth look at every function a user will be able to perform within the Kite app. Included also within this document is the purpose of this project and the scope of where the project hopes to be come Spring term and the Engineering Expo.

## CONTENTS

### REVISION HISTORY

Name	Date	Reason For Changes	Version
Project Requirements	Nov 1	Initial	Version 1

## I. INTRODUCTION

### *A. Purpose*

In the world filled with social media and online entertainment, lives are posted online 255 characters at a time, sometimes with a picture for good measure. Not one social media site is a complete solution for every person and thus leaves some without a way to express themselves online. This application aims to solve this by providing a more in-depth story driven posting experience. The purpose of the Kite project is to design and create this in-depth journal-like experience within the Kite app, a social media journaling community.

### *B. Project Scope*

\*See Problem Statement for Project Scope. (Reference 1)

### *C. Definitions, Acronyms, and Abbreviations*

#### **Acronyms**

- 1) **TBD:** To Be Determined

#### **Definitions**

- 1) **Kite:** Name of the project and related mobile app.
- 2) **Event:** A grouping of entries of a span of time i.e. "Vacation" or "Senior Year".
- 3) **Community:** A public profile that allows Kite users to post entries on a communal page.
- 4) **Tags:** TBD
- 5) **Moderator:** User with the permissions to edit and change a community's page.
- 6) **Admin:** User with the permissions to delete any user and post on the Kite application.
- 7) **Entries:** A single post from one user onto the Kite application.
- 8) **Journal:** A journal is a collection of a user's or community's entries.

### *D. References*

- 1) Problem Statement, 10/18/17, Group Kite, [Kite Problem Statement](#)

### E. Overview

This document aims to provide developers, managers, users and testers insight on how this project will be built and what this project is required to offer.

- Developer should read the Introduction section then focus on the Specific Requirements section as that will contain the specific points of interest and the list of requirements to finish the Kite project.
- Managers will want to continue their reading after the introduction section to the overall description where they will learn the overall structure of the project and all it entails.
- User should read the overview and table of contents to find the exact information they are looking for.
- Testers will need to view the Specific Requirements section as it will give them the most in-depth view into each individual function and feature within the Kite application.

## II. OVERALL DESCRIPTION

### A. Product Perspective

The project, originally called Social Mobile Journal App, was sponsored by David Vasquez, a developer at Oregon State University. This self-contained project was brought up as one of the proposed projects for Oregon State University's Senior Design Capstone course for Computer Science majors.

### B. Product Functions

#### 1) Database:

- **Required Tables:** The database must contain, at least, the required tables as provided by the client.

#### 2) Networking:

- **Server Side Security:** The application must protect against SQL injection
- **Secure Login/Logout:** The application must protect and secure the user's information when logging in or out.
- **Secure Username/Password:** The application must secure the username and password when retrieving from and saving to the database.

#### 3) App Design:

- **Android OS:** The application must be executable on the Android Operating System.
- **iPhone OS:** The application must be executable on the iPhone Operating System (iOS).
- **Modular:** The application must be modular in it's design, according to the client's review and approval.

#### 4) User Interface:

- **User Log In:** The user has the ability to log into their already created account.
- **User Log Out:** The user has the ability to log out of their account.
- **Post Entry:** The user has the ability to post an entry to their own journal. The user may also post to Community pages that they are apart of.

- **Create Community:** The user has the ability to create a Community, when created the user is set to be the moderator of that Community.
- **Create Event:** The user has the ability to create an Event which will allow them to group together multiple Journal Entries.
- **Edit User Profile:** The user has the ability to edit their own Profile.
- **Edit User Settings:** The user has the ability to edit their own settings.
- **Edit Community Profile:** The user has the ability to edit Community Profiles that they are moderators for.
- **Edit Community Settings:** The user has the ability to edit Community settings that they are moderators for.
- **Edit Entry:** Past Entries that a user has posted have the option to be edited.
- **Edit Events:** Past Events that a user has created have the option to be edited.
- **Follow Users:** A user has the ability to follow and unfollow other users.
- **Follow a Community:** A user has the ability to follow and unfollow Communities.
- **Follow an Event:** A user has the ability to follow and unfollow other user's Events.
- **Like an Entry:** A user has the option to Like other user's Entries.
- **Like an Event:** A user has the option to like other user's Events.
- **Like a Comment:** A user has the option to like other user's Comments.
- **Tag User:** When posting an Entry, Comment, or Reply, the user has the ability to tag other users.
- **Tag Entry:** When posting a Entry, the user has the ability to tag the Entry with descriptive words.
- **View User Profile:** A user has the option to view their own profile, or other user's profiles depending on the other user's privacy settings.
- **View Community Profile:** A user has the option to view any Community profile depending on that Community's privacy settings.
- **View Discover:** The Discover page is a randomly generated list of Entries that the user might be interested in.
- **View Timeline:** The Timeline page is a list of Entries that is sorted to have the most recent Entries to be first.
- **View User Setting:** A user has the ability to view their settings without entering the edit setting page.
- **View Community Setting:** A user has the ability to view the settings page for the Community, for which they are a moderator of, depending on that Community's privacy settings.
- **View Entry:** A user has the ability to view an individual Entry.
- **View Calender:** A user has the ability to view their Calendar page, which displays all Entries and Events according to the date they were posted.
- **View Event:** A user has the ability to view the Events page and view Events they created.
- **View Notifications:** A user has the ability to view notifications from a central tab/location.
- **Search:** A user has the ability to search for other users, Entries, Events, Communities, and tags.
- **Create Profile** A user has the ability to create an account.
- **Delete Profile:** A user has the ability to delete their account completely.

- **Disable Profile:** A user has the ability to disable their account publicly, but they have the option to re-enable their profile with their settings and Entries intact.

### *C. User Characteristics*

This application, being a social network, will have several tiers of administrative privileges. The highest level tier will be the Kite Administrators who have overall control of the application - they are there simply to moderate users (remove them, if need be) and assist users. The next tier down are the Community Administrators. Community Admins are the original creators of a Community or are the user given overall control of the page. These Administrators are the controllers of the Community's they've created; they are general users who have created a Community. They set the privacy and other restrictions on the Community and have the ability to give other users, who are apart of that Community, moderator privileges. These Community Moderators have much the same access and power as the Community Admin but without the ability to remove other Mods and the Admin. Below the Community Moderators are the Community Members. These Members have very little in the way of additional power but rather have joined a Community and have the ability to post entries and view others. The next and final tier down are the general users. These are users who have not created a community and have normal access to the application and it's functionality. They have control over their own privacy settings as well as their profile page.

### *D. Design and Implementation Constraints*

For this project, we'll be using React Native which will allow us to develop this application for both iOS and Android users. The application will have to be secure to keep user data and information safe. Due to the time restrictions of this project, some of our goals are listed as stretch goals - these are goals we hope to achieve, but are not considered mandatory for the completion of the project. An example of one of these goals would be the ability for a user to instant message other users.

### *E. Assumptions and Dependencies*

We are using MyPHP Admin for the database for our project. We are assuming it will be able to support the amount of users required for this project. We are depending on this system to be able to support all of our test users and actual users while we go through development and implementation of this project. Another possible problem we could run into is that we believe react native will work well for both iOS and Android operating systems (there could be issues in actual development that prevent certain functionality from working).

## III. SPECIFIC REQUIREMENTS

### *A. Database*

Within the database, there will be a required number of tables to fit the Client's specifications to complete the application. These tables consists of, but not limited to:

- Users
- Following

- Posts
- Notifications
- Settings
- Profile

### *B. Network and Security*

Security is essential to the Kite application as it will contain potentially hundreds, maybe even thousands, of user's private information. There are a few security measures required for this project to secure networking to and from the database and the Kite application. On the server side, SQL injection must be accounted for and disallowed to prevent database tampering. During login/logout and any other database accessing functionality, personal information and data must be securely transferred. Critical data will also not be stored as plane text in the database.

### *C. App Design*

The design of the Kite application will be executable on both Android and iPhone operating systems. In addition to being available on two major operating systems, Kite will be designed and coded in such a modular way that allows for easy understanding of its inner workings, quick access to specific sections of code, and few, if possible no, barriers to continued development after this project is completed.

### *D. User Interfaces*

- **Login / Create New Profile:** The Login / Create New Profile page is the first page a user will see and will provide the User with a way to create a new account or log into the application.
- **User Profile:** A User will have a User Profile, this profile is the location for a User to provide information about themselves.
- **Edit User Profile:** The Edit User Profile page will contain the same information as the View User Profile, but the difference is that this page will enable the user to edit the information in each section.
- **User Settings:** The User Settings page is only viewable by the corresponding User. In other words a User can not view other Users Setting page. The Settings Page displays all settings that the User has turned on and off.
- **Edit User Settings:** The Edit User Settings page is very similar to the View User Settings page. Only the corresponding user can edit their Settings. Once a user is in this page they have the ability to edit the setting within this page.
- **Discover:** The Discover page is a location for a User to view new Entries by other Users and Communities.
- **Create Entry:** The Create Entry page allows the User to create a new Entry.
- **Edit Entry:** The Edit Entry page is very similar to the Create Entry page, the difference is that the Edit Entry page will be pre-populated with the entries information.
- **View Entry:** The View Entry page will be like the Create Entry Page but it will not allow the User to change anything. This will also be the page where a User will navigate to if they click on another Users Entry.

- **Create Event:** A User has the ability to create an Event, an Event is a collection of Journal Entries.
- **Edit Event:** The Edit Event page will give the User the ability to click back onto old Events and be able to change the information about them.
- **View Event:** The View Event page will be the main page to see a Event, any User will be able to view the Events through the View Event Page.
- **Search Results:** There will be Search capabilities for the User, when a Search is finished it navigates the User to a new page with a collection of related Entries.
- **Community Profile:** A User has the ability to create a new Community and when they do they will be set to the main moderator for Community. A Community profile will contain information about the Community.
- **Edit Community Profile:** The Edit Community Profile page allows the Community Administrator of the Community to change the information about the Community.
- **Community Settings:** The Community settings page is viewable by all of the members of the Community, and displays the current setting.
- **Edit Community Settings:** The Edit Community settings page is where the Moderators are able to change the settings of that Community.
- **Create Community Entry:** The Create Community Entry page is similar to the Create Entry page, but when the Entry is created the Entry is connected to the Community and not the User that posted it.
- **View Community Entry:** The View Community Entry page will just be a page in which a user can see the Entries of a given Entry, and if they have the given permissions there will also be a edit button enabled for them to click on to navigate them to the edit Community Entry page.
- **Edit Community Entry:** The Edit Entry page will allow the moderators or the User that initially posted it to be able to edit the Entry and change the values in all of the fields.
- **Community Timeline:** The Community Timeline is a page similar to the User's Timeline, but this Timeline is viewable by all Users depending on if the moderators have given that permission. The Timeline will only consist of Entries from the Community.

#### *E. Hardware Interfaces*

This software application will be supported on both iOS and Android systems. It will utilize the touch screen interfaces of both mobile devices as well as their hardware buttons to operate.

#### *F. Software Interfaces*

This application will have a mobile interface that sends and receives data from a database. The user will be able to secure login to the application, make posts that will be saved in the database as well as make comments and replies, which are also saved.

#### *G. Communications Interfaces*

This application will have email communication with the user in case of lost passwords and as account validation. It will also have a network connect with a database to send and retrieve data. The user can submit an

entry (post) which other users who are following the original user can see. All users can make comments on posts as well as reply to these comments.

#### *H. Stretch Goals*

As a group consensus, these stretch goals are features that would be beneficial to the overall goals of the project but are not feasible within the time restraints. After the project has completed and progressed past its primary goals, then these stretch goals will be the next steps for the project.

- Database Backup
- Non-Infinite Timeline Scrolling
- Photo Filters
- UI Layout Configurations
- Discover pages
- Location-based queries
- Video Integration

## IV. INDEX

#### *A. Safety Requirements*

This product will not cause physical harm or damage to the user. However, there are security measures it will take (See 'Security Requirements').

#### *B. Security Requirements*

This application will have secure login, registration, preventative measures against SQL injection, and no critical plain text entries stored in the database. Users will also have the ability to enable different privacy settings to ensure they can prevent other users from viewing their data and personal information.

#### *C. Software Quality Attributes*

This application will be flexible in the sense that it can run on both iOS and Android. In this sense, both an Android and iOS user will be able to use the application and all functionality equally. The application will be testable - developers will be able to run test posts, other user functions, and verify their success through the database.

#### *D. Business Rules*

##### **User Permissions Hierarchy:**

##### **• Kite Administrator:**

A Kite Administrator has complete control of the entire app. These users will be the Kite designers, overseeing managers, quality control members, and the almighty Kite creators.

##### **• Community Administrator:**

The Community Administrators are the users who create a Community and have the power to control it and its



user's input. The Community Administrator has the power to give Community Members moderator privileges. The Community Administrator can delete entries and add/remove users into the Community.

- **Community Moderator:**

Community Moderators are given privileges by the Community Administrator of a certain Community they are apart of. These Moderators only have as much power as the Community Administrator gives them.

- **Community Member:**

A Community Member is a member of a Community.

- **User:**

The general user of the Kite app.

