LetsGo

Requirements Specification

Version 1.0

February 10, 2020

–

Suchi Kapur

Nick Garrett

Khaled Khalil

Table of Contents

1. Introduction

1.1 Project Overview

LetsGo is a community-based application that embraces groups with similar qualities and attributes, and allows users to create communities and events for members that align themselves with said qualities, attributes and interests. Events that are created within communities of the application aim to further integrate users with one another, embracing their shared ideals and interests. LetsGo focuses on inclusion of users rather than exclusion; if a user does not fit into a community, there will be others they can join, or they can create their own. Everyone has a place within the LetsGo application, and no user will be left alone. LetsGo can be used by people ages 12 and older of various backgrounds, and is targeted towards people seeking social interaction based on similar interests. Specifically, LetsGo is primarily focused towards social groups larger than 10 people that require more coordination, leadership, and structure within organizing events. The purpose of the LetsGo is to connect users with similar interests, where users will be able to find their niche, or place, within the communities of the application.

This social media application is influenced by the main features of other social media platforms such as Facebook Invite, but will address concerns of similar preexisting applications regarding ease of use, as well as expanding on and creating new features that interest users. LetsGo combines social media features with event planning to create a user-friendly, exclusive application to promote community engagement and individual expression. The major aspects of the LetsGo application include user profiles, communities, and public and/or private events. The social media features of LetsGo include community formation and the ability for users to join, lead, and create groups based on a similar set of interests, or interest tags.

1.2 Definitions, Acronyms, and Abbreviations

Term	**Definition**
Stakeholder	Any individual that influences the development and progress of the project.
User	Any person that uses the application.
Community	Groups of people formed in the application with the same interest(s).
Community Leader	Has authority to remove members from their community, delete public community events and delegate privileges to community members.

Event	A planned and organized occasion created by a user.
Announcement	Only community leaders have access to announcements and are used to help clarify event plans or changes to the community alignment
RSVP	A request to accept or decline an invitation to an event.
Interest/Interest Tag	A tag that defines a user’s individual expression and create their identity
SRS	Software Requirements Specification
Xamarin	An application platform for building iOS and Android compatible apps.
Visual Studio	Development platform.
Firebase	Database used for back-end. Used to store information pertaining to users and user accounts.

1.3 Purpose, Scope, and Organization of this Document

The purpose of this SRS is to detail the functionality and requirements of the LetsGo mobile application. The document will also explain the various application features and provide overall descriptions, objectives, functions, assumptions, dependencies, system constraints and requirements for the development of LetsGo. The intended audiences of this document are the stakeholders of the LetsGo application and LetsGo developers.

LetsGo is not intended to replace mainstream social platforms, but aims to integrate social media features with event planning and organization for users in a central application. Additionally, social aspects of LetsGo are not intended for information sharing, but rather expression of interests and guiding social gatherings for better community interaction. The intended purpose of this application is to enhance event planning, promote community engagement, and embrace individual expression.

The requirements laid out in this document contain functional and non-functional requirements for the LetsGo application, as well as outlining and explaining interfaces and any other requirements that are important to the development of the LetsGo application.

2. Overall Description

2.1 Product Environment

LetsGo is a mobile social media application, offering communication to others through the internet. LetsGo is an independent stand-alone application that is similar to social media applications such as Twitter and Facebook. LetsGo will utilize internet connection and a real-time database (Firebase) for communication and storing relevant application and account information such as user profiles, events, and communities (see Figure 1 below).

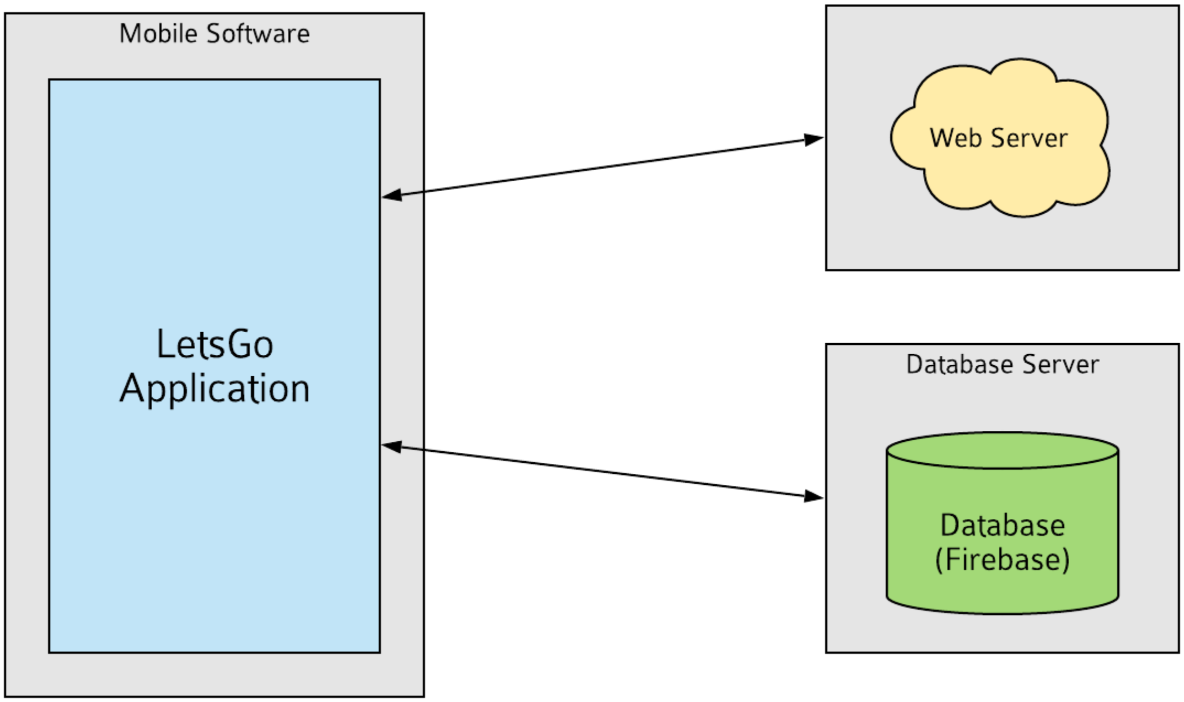


Figure 1. Diagram of Components, Relationships, and External Interfaces.

System Interfaces

LetsGo is a mobile application that connects to the internet, or a web server, to allow users to connect within communities, communicate, and use all features of the application. The application will also utilize a database server (Firebase) to store information pertaining to user accounts, user profiles, communities, and events. Users will not have direct access to this database server, but will be able to access available information through the use of the application.

User Interfaces

The user interface utilizes the touch screen interface of the mobile device, where the user can press icons on the screen to create an account (for first-time users), login, manage events, manage communities, manage friends, manage their accounts, view their feed, chat, search, and logout. User interfaces will be implemented for user-based functions within the application (see SRS Section 3.1). The use case diagram below illustrates the various user interfaces and user-based functions (Figure 2). User interfaces will be formatted to device screens in portrait layout.

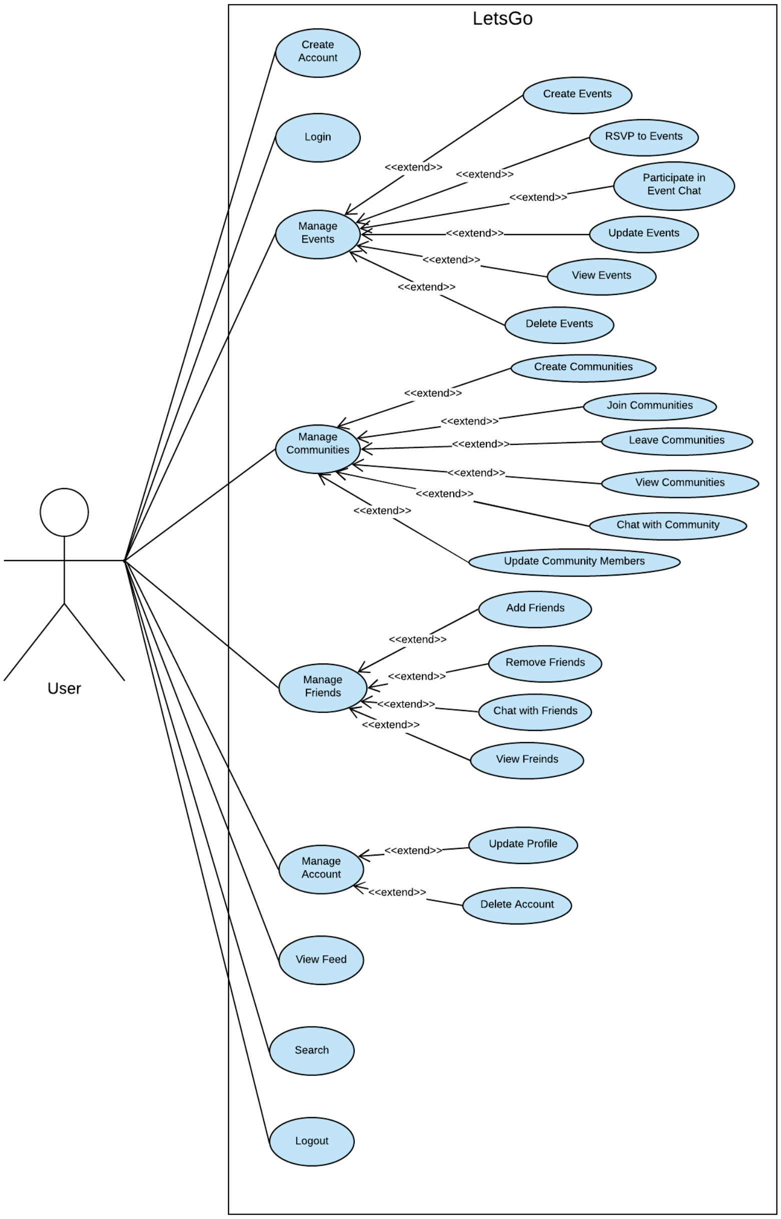


Figure 2. Use Case Diagram for User Interfaces.

Hardware Interfaces

The hardware interface for the LetsGo application are mobile devices (smartphones). To use the LetsGo application, a device that supports iOS or Android applications must be used.

Software Interfaces

For the development of the LetsGo application, Xamarin will be used. User mobile devices utilizing an iOS platform require version 13.3 or later, and those utilizing Android-based applications require version Pie 9.0 or later.

Communication Interfaces

For direct or group/community messaging, XMPP (eXtensible Messaging and Presence Protocol) will be needed to handle a message delivery system. The chat interfaces will also utilize a database server connection to store messages. The LetsGo application will also utilize a connection to a web server (the internet) for usage of certain application features.

2.2 User Characteristics

People ages 12 and older who are seeking social interaction based on similar interests are the target audience, and therefore users, of the LetsGo application. Users of the LetsGo application should have previous knowledge and experience of using mobile applications, using and having social media accounts, and navigating through various pages within a mobile application.

The users will have the ability to join communities that share similar interests and values or create their own, join or create events for their communities, and connect with other users with similar interest tags through a search feature.

Users of LetsGo will need to create an account that provides information such as their email, password, first and last name, and date of birth. Once a user has an account, they will be able to create or update their profiles with their interests, hobbies, communities, their location (city) and profile picture.

2.3 Product Functions

Once downloaded, users of the LetsGo application will be directed to a login page. First-time users will be able to create an account. Users that have accounts already will be able to login to the application and view their public feed (homepage) that displays location-based public events. The application will allow users to navigate to numerous pages such as their own profiles, viewing and managing friends, events, and public feeds. Users will also be able to search for users, events, and communities based on interest tags.

Function: Create Account

First-time users will have the ability to create an account for the LetsGo application. When a user creates an account, they will also create their user profiles.

Function: Login

When opening the app, if a user is signed out of their LetsGo account, they will be directed to a login page and prompted to login. The login page will contain fields for the user to enter their email address and password to login. The login page should include a "Forgot Password" feature for users to reset their passwords.

Function: Manage Account

Users will have the ability to manage their accounts by updating their profiles. User profiles include their full names, date of birth (cannot be changed), interests, hobbies, location (city), and a profile picture. Users will also be able to delete their accounts.

Function: Manage Friends

Users will be able to view their friend list in a single page, and are given an option to "unfriend" if they choose to do so. Users will also be able to click any friend on this page to view their profiles. Users will be able to add friends by searching for interest tags or by email address and sending a request. Users will also be able to chat in a direct message with friends.

Function: Manage Communities

Users will be able to view communities that they are a member of, view specific community pages, create communities based on interest tags, leave communities, view community announcements, and join communities. Users will be able to participate in (group) chats with communities that they are members of. Users will be able to view specific community identities and community pages.

Function: Manage Events

Under this section, users will be able to view events that they have accepted an RSVP to. Users will also be able to accept or decline any event invitations, create events, and view/participate in event group chats.

Function: View Feed

Users will be able to view public events in their specified location (city) on their public feed. The feed will appear on a single page, where the user can view and scroll through public events in their city.

Function: Search

Users will be able to conduct searches through a search bar. Searches are queried by interest tags associated with users, communities and events. In the search query, only public users, communities, and events will appear.

Function: Logout

The user will be able to log out of the LetsGo application.

2.4 Assumptions and Dependencies

Being a mobile application, LetsGo users must use a mobile device that supports iOS or Android mobile applications. Users of the application must also have an internet connection through their cellular service provider or internet service provider for certain features of the application to work. Users must have an account, or create one if they do not have one, to use any and all of the features of the application.

2.5 Constraints

As the LetsGo application is for mobile devices, users are restricted to the usage of the application on only mobile devices. To use certain features of the LetsGo application, users will also need to have an internet connection.

Users will be restricted to viewing only the aspects of the application that are available on their screens. All user information, events, tags, etc. will be stored in a database that is out of the user's reach. Users will only have access to the information within the database through the application's functionality for security purposes.

The LetsGo application will be developed using Xamarin on Visual Studio, an application platform used for building iOS and Android applications. Information from user accounts and of the LetsGo application will be stored in a database (Firebase).

3. Requirements

This section will discuss user interface, functional and non-functional quality requirements of the LetsGo application.

3.1 User Interface Requirements

3.1.1 Create Account Interface

The “Create Account” function allows first-time users to create an account for the system.

To create an account, users will be prompted to enter their first and last name, date of birth, email address, and password.

When a user creates an account, they will also create their user profiles by providing their interests, hobbies, values, location (city) and a profile picture. Users can set their profiles to “public” or “private.”

Source of Input OR Destination of Output	createAccount.xaml createAccount.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	New user accounts will be added to the Firebase database. After creating an account, users will be redirected to their user profiles (see Section 3.1.4 for managing accounts).
Menu Structures	The “Create Account” option will appear on the login screen.
Error/Other Messages	1. You have successfully created an account! Please update your profile. 2. Account creation was unsuccessful. Please try again.
Function Keys	“Create Account” button. “Next: Update Profile” button.

3.1.2 Login Interface

The “Login” function of the LetsGo application allows users to login to the system.

The login page will display the LetsGo logo, and will contain fields for the user to enter their email address, password, and a login button.

Source of Input OR Destination of Output	login.xaml login.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	Login status will be updated in the database.
Menu Structures	Users will be able to login upon opening the application on the login screen (if they are not signed in already).

Error/Other Messages	1. Successful login. 2. Invalid login credentials. Please try again.
Function Keys	“Login” button

3.1.3 Forgot Password? Interface

The “Forgot Password?” function of the LetsGo application allows users to reset their passwords.

The “Forgot Password?” button will be on the login screen.

Source of Input OR Destination of Output	forgotPassword.xaml forgotPassword.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	User passwords will be updated in the Firebase database.
Menu Structures	The “Forgot Password?” function will appear on the login screen of the application.
Error/Other Messages	1. Your password has been reset. Please log in. 2. Your password has not been reset. Please try again.
Function Keys	“Forgot Password?” button/link

3.1.4 Manage Account Interface

The “Manage Account” function of the LetsGo application allows users to create or edit their profiles, or delete their accounts.

When updating profiles, users can update their full names, interests, hobbies, location (city), “public” or “private” account status, and profile pictures. Users will not be able to change their date of birth.

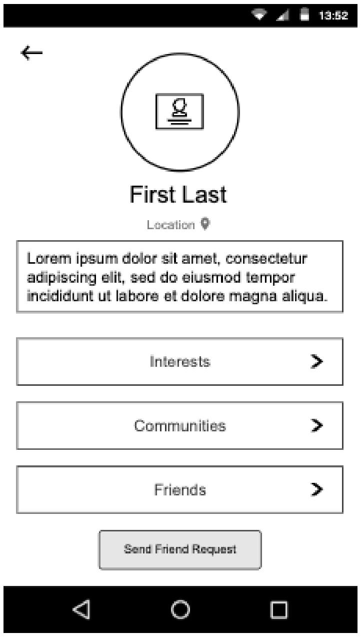
Source of Input OR Destination of Output	userProfile.xaml deleteAccount.xaml userProfile.xaml.cs deleteAccount.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application

Report Layouts	User profile/account information will be updated in the Firebase database.
Menu Structures	The “Manage Account” function will appear on the user’s profile as well as a 3-lined menu icon. The user’s profile will also be an icon on the navigation bar at the bottom of the application’s screen.
Error/Other Messages	1. Profile was updated! 2. Profile has not been updated. 3. Your account has been deleted.
Function Keys	1. “Edit Profile” button 2. “Delete Account” button 3. “User Profile” icon

3.1.5 Manage Friends Interface

The “Manage Friends” function of the LetsGo application allows users to add friends, remove (unfriend) friends, chat with friends, and view their friends and friends’ profiles.

Source of Input OR Destination of Output	viewFriends.xaml manageFriends.xaml chatFriends.xaml viewFriends.xaml.cs manageFriends.xaml.cs chatFriends.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	User friends will be added or removed from the database.
Menu Structures	The “Manage Friends” function will appear under a 3-lined menu icon.
Error/Other Messages	1. Users are friends! 2. Users are not friends. 3. User has been unfriended. 4. You are chatting with your friend.
Function Keys	1. Pressing on a friend’s profile will allow the user to view their friend’s profile (see Figure 3 for mockup). 2. “Unfriend” button 3. “Add Friend” button 4. “Chat with Friend” button



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Figure 3. Viewing a User’s Profile Screen (Users are not Friends).

3.1.6 Manage Communities Interface

The “Manage Communities” function of the LetsGo application allows users to join communities, leave communities, view communities they are members of, and chat with their communities. Users will only have access to viewing community pages that they are members of.

Communities will be listed on a single, continuous-scroll page. The user should be able to click onto any community to view their values and identity. Users will be able to view announcements made by community leaders.

Users should also be able to leave communities and view other members of each community that they have membership to. When a user leaves a community, if the community is “private,” the user must request the community leader to join again.

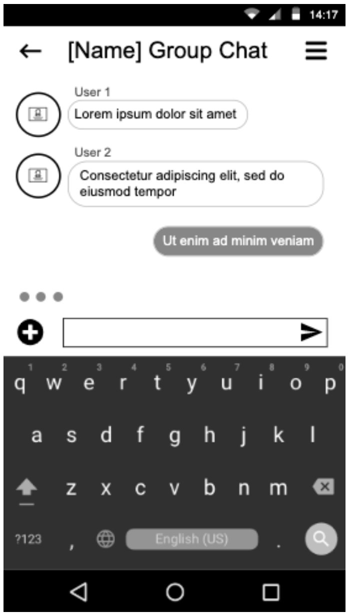
Users will also be able to request to join communities that they are interested in.

When creating communities, users will be able to enter the community name, community identity and interest tags, and community owner (the user will be the community owner/community leader). It is accepted that regular community members are subordinates of the community leader.

Users who are community leaders will be able to accept others into their community or remove members (community leaders will be able to update community members). Community leaders will also be able to delete public community events.

Source of Input OR	viewCommunities.xaml

Destination of Output	manageCommunties.xaml chatCommunities.xaml viewCommunities.xaml.cs manageCommunities.xaml.cs chatCommunities.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	Communities will be added or removed from the user in the Firebase database.
Menu Structures	The “Manage Communities” function will appear under a 3-lined menu icon.
Error/Other Messages	1. You have joined a new community. 2. You have not joined a new community. 3. You have left a community. 4. You are chatting with your community.
Function Keys	1. If the user selects a community from their “Communities” page, they shall be redirected to the community entrance. 2. “Enter Community” button 3. “Join Community” button 4. “Leave Community” button 5. “Chat with Community” button



[Name] Group Chat

User 1
Lorem ipsum dolor sit amet

User 2
Consectetur adipiscing elit, sed do eiusmod tempor

Ut enim ad minim veniam

Figure 4. Group/Community Chat Screen.

3.1.7 Manage Events Interface

The “Manage Events” function of the LetsGo application allows users to RSVP to events, create events, view communities that they have accepted invitations to, and chat within event pages.

To create an event, users shall provide the event name, event date, event time, location, and event description. When creating an event, users will also be able to add an image. Events can be made to “public” or private,” where public events can be seen by all users regardless of community association or interests, and private events can only be seen by users who have been invited. Event owners will be able to invite friends/users to events.

Event owners shall be able to edit or update event information, or delete events. Events that are deleted will be removed from the database.

Source of Input OR Destination of Output	viewEvents.xaml manageEvents.xaml chatEvents.xaml viewEvents.xaml.cs manageEvents.xaml.cs chatEvents.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	Events will be added or removed from the user in the Firebase database.
Menu Structures	The “Manage Events” function will appear under a 3-lined menu icon.
Error/Other Messages	1. You have created a new event. 2. Event creation unsuccessful. 3. You have deleted an event. 4. You have successfully updated an event. 5. Event update unsuccessful. 6. You have submitted an RSVP to an event. 7. RSVP submission unsuccessful. 8. You are chatting with event-goers of this event.
Function Keys	1. “Create Event” button (see Figure 5 for event creation mockup) 2. “Delete Event” button 3. “Update Event” button 4. “RSVP to Event” button 5. “Chat with Event-Goers” button

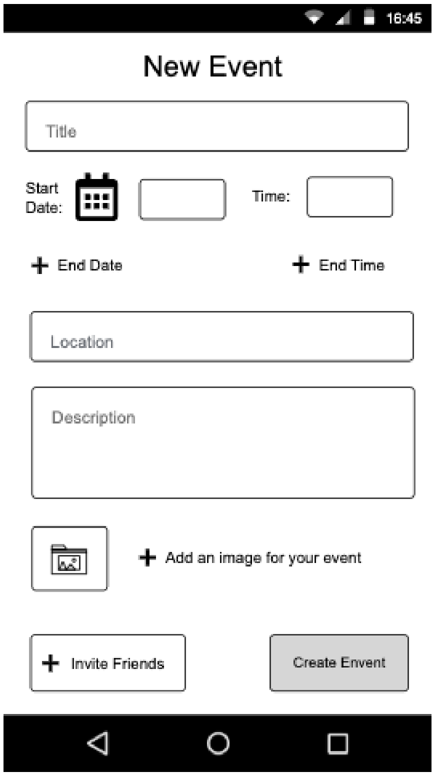


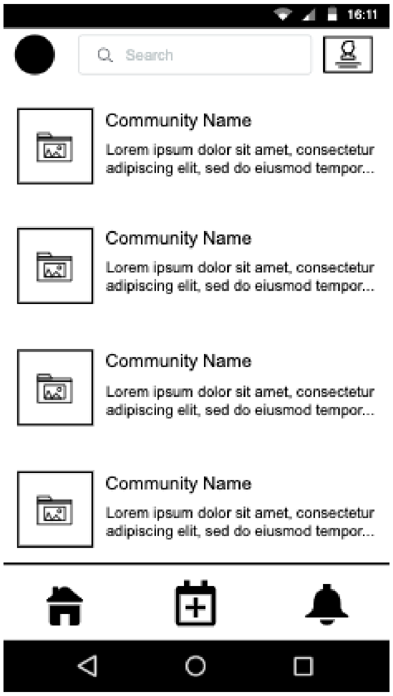
Figure 5. Event Creation Form.

3.1.8 View Feed Interface

The “View Feed” function allows users to view public events in their location on a single, continuous-scroll page. Events are displayed with the community the event was created from, a description of the event, and an image.

Source of Input OR Destination of Output	viewFeed.xaml viewFeed.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	Events for feed-viewing will be retrieved from the Firebase database for this function. The user’s feed will appear in a single, continuous-scroll page.
Menu Structures	The “View Feed” function will be the homepage of the app. The “View Feed” function will also appear on the navigation bar at the bottom of the application screen.
Error/Other Messages	1. You are viewing events in your location. 2. Error loading feed.

	3. There are no events in your location.
Function Keys	“Feed” icon.



16:11

Search

Community Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor...

Community Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor...

Community Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor...

Community Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor...

Figure 6. Home Feed Screen Showing Public Events in Location Based on Community.

3.1.9 Search Interface

The “Search” function allows users to search for public users, events, and communities. Searches are conducted by interest tag or email (for searching a specific user).

Source of Input OR Destination of Output	search.xaml search.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	The “Search” function retrieves information from the Firebase database with matching interest tags, or email.
Menu Structures	The “Search” function will be at the top of the application feed, appearing as a search bar.
Error/Other Messages	1. Results matching your search: 2. No results found.

Function Keys	"Search" bar and button.

3.1.10 Logout Interface

The "Logout" function allows a user to logout of the system.

Once logged out, the user will be redirected to the login screen/page.

Source of Input OR Destination of Output	logout.xaml logout.xaml.cs
Required Screen Formats/Organization	Android Mobile Application iOS Mobile Application
Report Layouts	The "Logout" function updates the login status of the user in the Firebase database.
Menu Structures	The "Logout" function will appear under a 3-lined menu icon.
Error/Other Messages	1. You have logged out.
Function Keys	"Logout" button.

3.2 Functional Requirements

Functional requirements of the LetsGo application are detailed below and organized by feature. Requirements are prioritized as follows:

- Priority 1 – The requirement is a "must have" as outlined by policy/law
- Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
- Priority 3 – The requirement is a "nice to have" which may include new functionality

3.2.1 Create Account Requirements

Req. No.	Requirement	Priority	Comments
FR_CR_1	The system shall allow users to enter their first name, last name, date of birth, email address, and password in a form to create their account. (This information will be used to create a new user in the database).	1	
FR_CR_2	The system shall perform validity checks to ensure that	1	

	email addresses provided by the user are not in use. If the email address is already in use, the user will be prompted to enter a different email or login with an existing account.		
FR_CR_3	The system shall validate that the user age is at least 12 years old. If the user does not meet the age requirement, they will not be able to create an account.	1	
FR_CR_4	The system shall authenticate that user-provided passwords meet specified password requirements. (See Section 3.3.1 for password requirement details).	1	
FR_CR_5	The system shall allow the user to login to their new account after account creation.	1	

3.2.2 Login Requirements

Req. No.	Requirement	Priority	Comments
FR_LI_1	The system shall present/call a login form for the user to enter login credentials.	1	
FR_LI_2	The system shall authenticate user login credentials. If user login credentials are invalid, the system shall prompt the user to retry, recover their password (“Forgot Password?” link), or create a new account.	1	
FR_LI_3	The system shall provide the user with a “Forgot Password?” option. The “Forgot Password?” option will allow the user to reset their password after verifying account information.	2	
FR_LI_4	The system shall redirect the user to the LetsGo homepage after successful login.	1	

3.2.3 Password Reset Requirements

Req. No.	Requirement	Priority	Comments
FR_FP_1	The system shall allow the user to recover/reset their	1	

	password. The user will receive an email with a link to reset their password.		
FR_FP_2	After the user clicks the reset password link in their email, the system shall allow the user to enter their new password and confirm their new password. The user’s new password must meet password requirements (See Section 3.3.1 for password requirement details).	1	
FR_FP_3	The system shall allow the user to login with their new password after a successful password reset. The new password will be updated in the database.	1	

3.2.4 Manage Account Requirements

Req. No.	Requirement	Priority	Comments
FR_MA_1	The system shall allow the user to edit/update their user profile (full name, interests, hobbies, values, location/city, profile picture) through an update profile form. All updated information in a user’s profile will be updated in the database.	1	
FR_MA_2	The system shall allow the user to delete their account. The corresponding account information will be removed from the database.	1	

3.2.5 Manage Friends Requirements

Req. No.	Requirement	Priority	Comments
FR_MF_1	The system shall display the user’s friends in a single page when the user clicks on their “Friends” tab/icon.	1	
FR_MF_2	The system shall give the users the option to “unfriend” any friends by clicking a button/icon. The user will be prompted with a confirmation dialogue box to confirm unfriending a user. Users that have been unfriended will be reflected and updated in the database.	1	
FR_MF_3	The system shall allow users to click on any friend on	1	See Figure 3 for

	the “Friends” page to view the desired user’s profile.		viewing a user’s profile.
FR_MF_4	The system shall allow users to add friends by sending a “friend request.” Friend requests can be accepted or denied by the receiving user. Accepted friend requests will reflect both users having a new friend in the database.	1	
FR_MF_5	The system shall allow the user to chat with a friend. The user will click on the desired friend to chat with, and the system will create the chat connection for direct messaging to occur.	2	

3.2.6 Manage Communities Requirements

Req. No.	Requirement	Priority	Comments
FR_MC_1	The system shall allow the user to view communities that they are members of through their “Communities” page.	1	
FR_MC_2	The system shall redirect the user to the community entrance that displays the identity of the specified community, if the user clicks on a community on their “Communities” page.	1	
FR_MC_3	The system shall redirect the user to the community page that shows community events and community chats if the user selects the “Enter Community” button.	1	
FR_MC_4	The system shall allow users to create communities by using a form with fields of community name, interest tags, and community owner/leader (which is the user creating the community). The new community will be created in the database, with the owner as a member.	1	
FR_MC_5	The system shall allow users to join communities. Users will be able to immediately join communities that are specified as “public,” or send a request to those that are listed as “private.” The community leader must accept requests to join. New communities that a user becomes a member of are updated and reflected in the database.	1	

FR_MC_6	The system shall allow users to leave communities by selecting a “Leave Community” button/option. When the user leaves a community, it will be updated in the database.	1	
FR_MC_7	The system shall update the community leader to be the member with the longest membership if the community leader leaves the community.	1	
FR_MC_8	The system shall allow community leaders to accept or deny user requests to join their community. Accepted members will be updated in the database.	1	
FR_MC_9	The system shall allow community members to participate in community group chats.	2	See Figure 4 for Group Chat Screens.

3.2.7 Manage Events Requirements

Req. No.	Requirement	Priority	Comments
FR_ME_1	The system shall allow users to view events that they have RSVP’d to on their “Events” page. If a user clicks on a certain event, the system shall redirect the user to view the event’s full details.	1	
FR_ME_2	The system shall allow users to create events by using and filling a form containing event name, event date, event start and end time, event location, an image, the option of the event being public or private, and the option to invite friends. New events will be created in the database.	1	Input fields for event information. See Figure 5 for Event Creation Form.
FR_ME_3	The system shall allow users to RSVP to events. If a user accepts an invitation, the event will be updated with a new event-goer in the database.	1	
FR_ME_4	The system shall allow event owners/creators to update name, date, time, location, and public/private status for events they have created. All updated information will be updated in the database.	1	
FR_ME_5	The system shall allow users to participate in event	2	See Figure 4 for

	chats for desired events. The system will create the connection for the chat.		Group Chat Screens.

3.2.8 *View Feed Requirements*

Req. No.	Requirement	Priority	Comments
FR_VF_1	The system shall display public events within a user's location (city) on their "Feed" page that the user shall be able to scroll through.	1	
FR_VF_2	The system shall refresh the user's feed if the user swipes down and refreshes the page.	3	

3.2.9 *Search Requirements*

Req. No.	Requirement	Priority	Comments
FR_SE_1	The system shall query the database when the user conducts a search. Search items that appear include public user profiles, public events, and public communities.	1	
FR_SE_2	The system shall allow users to select a desired search result to view the community page, event page, or public user profile.	1	

3.2.10 *Logout Requirements*

Req. No.	Requirement	Priority	Comments
FR_LO_1	The system shall allow users to log out of the system. Users will logout by selecting the "Logout" button.	1	
FR_LO_2	The system shall redirect the user to the login page upon successful logout.	1	

FR_LO_3	The system shall end the current application session upon unsuccessful logout, if the system crashes.	1	

3.2.11 Additional Requirements

Req. No.	Requirement	Priority	Comments
FR_AR_1	The system shall log a user out after 10 minutes of inactivity.	3	

3.3 Quality Requirement: Security

Requirements specified under this section protect data of the LetsGo application from unauthorized access and external threats.

3.3.1 Password Requirements

User passwords must meet the following requirements:

- ❖ Password must be at least 8 characters in length.
- ❖ Password must contain one number.
- ❖ Password must contain one uppercase letter.

Related Functional Requirements: FR_CR_4, FR_FP_2

3.3.2 Password Encryption

User passwords should be encrypted to prevent security issues, hacking of accounts, and potential external threats to the system's integrity.

3.4 Quality Requirement: Reliability

LetsGo should operate on a consistent basis (everyday) throughout the year. In optimal cases, the application should operate 24 hours a day, 7 days a week. The system will maintain data integrity at all times.

3.5 Quality Requirement: Availability

The LetsGo application will be available at all times, except in the cases of database or network/internet connection failure.

In the event of system, network, or database failure, availability is estimated to be zero until updates occur to ensure a functioning and secure application.

4. Requirements Confirmation/Stakeholder sign-off

Meeting Date	**Attendees (name and role)**	**Comments**
2/10/2020	Suchi Kapur, LetsGo Developer Nick Garrett, LetsGo Developer Khaled Khalil, LetsGo Developer	Discussed & Created SRS Sections 1.1-1.3
2/12/2020	Suchi Kapur, LetsGo Developer Nick Garrett, LetsGo Developer Khaled Khalil, LetsGo Developer	Confirmed SRS Sections 1.1-1.3 Discussed & Confirmed SRS Sections 2.1-2.5
2/14/2020	Suchi Kapur, LetsGo Developer Nick Garrett, LetsGo Developer Khaled Khalil, LetsGo Developer	Discussed & Confirmed SRS Sections 3.1-3.5
2/18/2020	Suchi Kapur, LetsGo Developer Nick Garrett, LetsGo Developer Khaled Khalil, LetsGo Developer Daniel Medina, Stakeholder	Stakeholder Sign-Off x________________ Suchi Kapur x Nick Garrett Nicholas Garrett x K K Khaled Khalil x Daniel E Medina Daniel Medina

References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.