



MealUp

Product Guide

Andrew Zeng
Annie Zou
May Jiang
Chi Yu

User Guide

Purpose

MealUp is a mobile app for iOS & Android that helps Princeton students schedule meals with their friends. Each user inputs their availability for meals over the next week and the app allows meal scheduling between users through a request system. Users can easily keep track of who they plan to eat with for the week and request & reschedule meals with friends & groups of friends through a user-friendly interface, and the app finds a time that works for everyone by matching the users' available times. The app supports Facebook login and imports Facebook friends for the user's convenience.

Installation

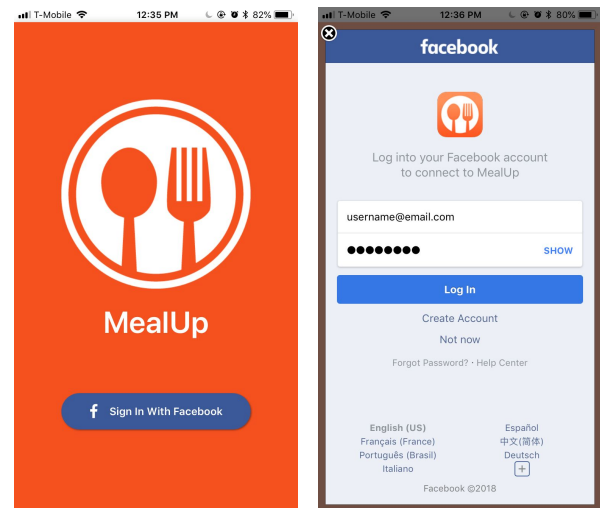
MealUp is available for download on the App Store for iOS (versions 9.0 and later) and the Google Play store for Android (versions 4.4 and later).

<https://itunes.apple.com/us/app/mealup/id1375233611?mt=8>

<https://play.google.com/store/apps/details?id=com.mealupteam.mealup>

Login

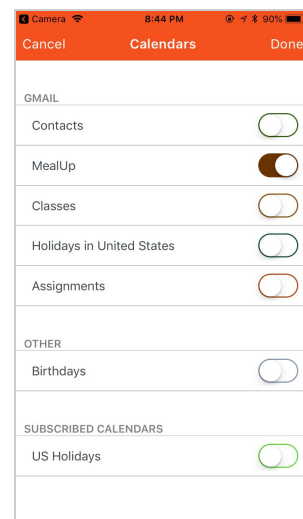
MealUp uses Facebook log-in for registration. To connect to MealUp, press the "Sign In With Facebook" button and sign in to your Facebook account. You will only be prompted this one time to sign in for initial registration, provided the app remains installed on your device. The first time you login, the app will guide you to swipe through a series of onboarding screens describing the basics of how to use the app.

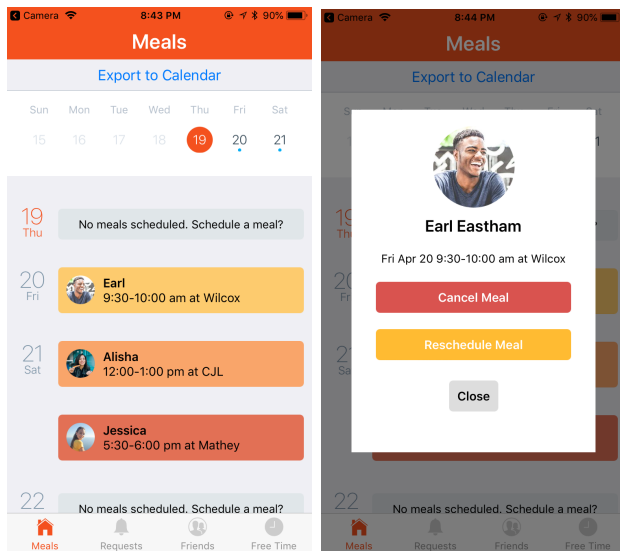


Home Screen

I. Export to calendar

At the top of screen below the orange navbar you can click on "Export to Calendar" and select the calendar you want to export your meals to; once you have selected a calendar, any future scheduled meals will automatically be added to that calendar. Note that you can only select one calendar.





II. Scheduled meals

The schedule that composes the majority of the homepage displays a list of all scheduled meals with friends and groups for the upcoming week (7 days). Each entry in the schedule contains information about the meal, namely the friend or group name, time of day, and location. Entries are color-coded by the type of meal based on the time of the meal, e.g. breakfast, lunch, dinner. Simply click on an entry to view meal information, or to cancel or reschedule the meal.

III. Navigation

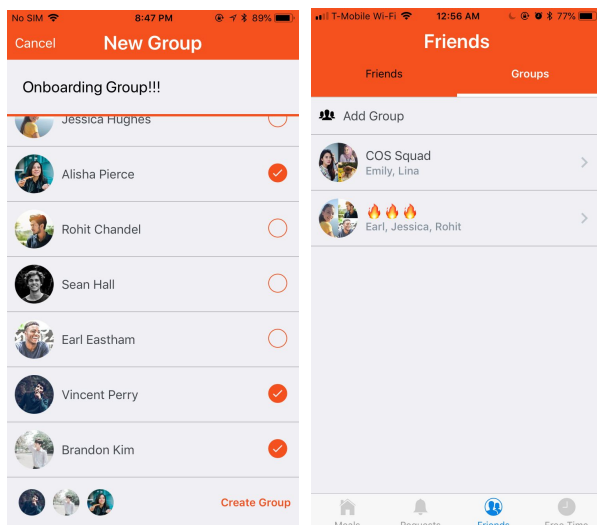
The main navigation of MealUp is controlled by the tab bar at the bottom of the screen. The first tab navigated to upon initial login is the fourth tab, the Free Time tab, to input availability for meals. On future sessions, the first tab navigated to is the homepage described above. The current active tab is indicated by the orange highlighted icon. From left to right, the tabs are Meals (homepage), Requests, Friends, and Free Time.

Friends Screen

The third tab in the tab bar takes you to the Friends screen. The Friends screen displays your friends that are on MealUp and the groups you are currently in.

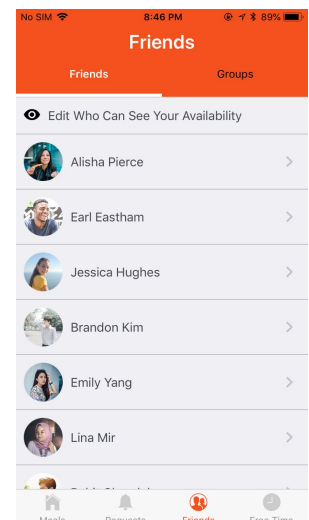
I. Friends

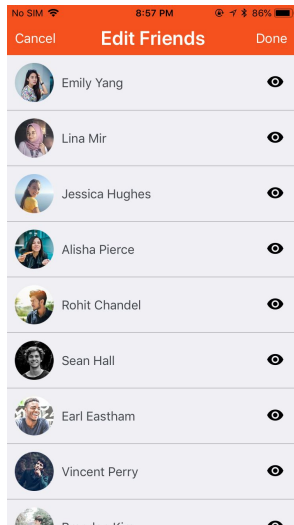
MealUp automatically imports your Facebook friends who have signed into the app, so you can easily find your friends and request meals by clicking on a friend. Friends are listed alphabetically.



II. Groups

To connect to friends more, users can form groups. To create a new group, click on "Add Group" on the Friends screen, select multiple friends, and optionally name the group. Any member of the group can then make group meal requests by clicking on a group. Any member of the group can also leave, rename, or add members to the group by long pressing on the group.





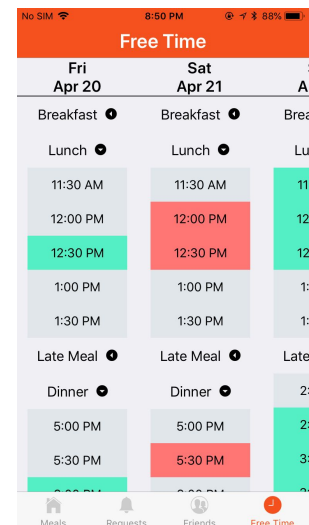
III. Privacy Settings

Users have control over which friends can access their matching freetimes and make meal requests. To access privacy settings, click on “Edit Who Can See Your Availability” at the top of the Friends screen. Click on a friend to turn your visibility to them on or off; your visibility to each friend is denoted by the eye icon next to his/her name.

Free Time Screen

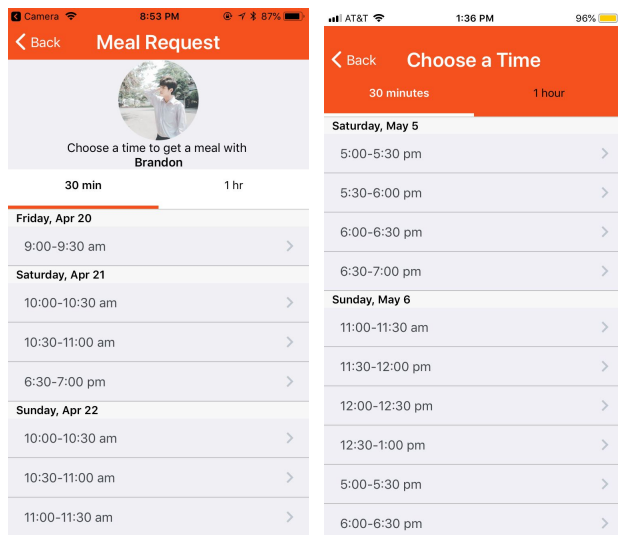
I. Setting Free Times

Users can select their free times by clicking on half-hour time blocks over the next 7 days on the Free Time Screen. The app automatically blocks out times that have passed and times where the user has already scheduled a meal in red, and defaults the next week’s data to the current week’s data so that users do not have to input times repeatedly from week to week.



Requests Screen & Making Meal Requests

The second tab takes you to the Requests screen. The Requests screen displays all requests you have sent and received. These requests expire automatically if the requested time for the meal has passed. To make a new request, click the plus icon at the top right of the screen.



I. Request by Friend

To grab a meal with a specific friend, select the friend and click on one of the times listed that both of you are available for.

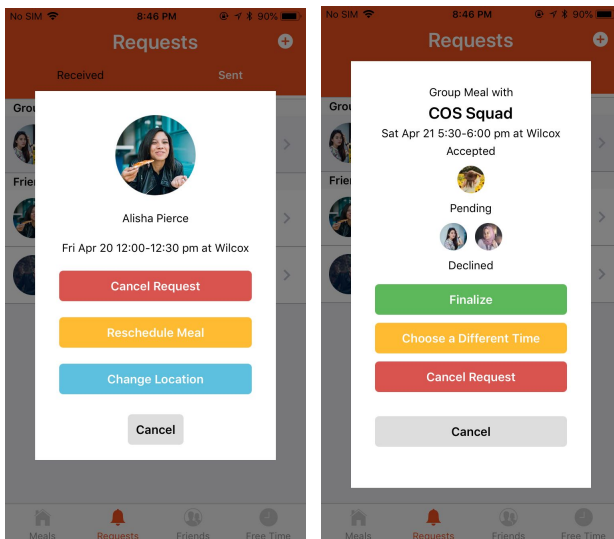
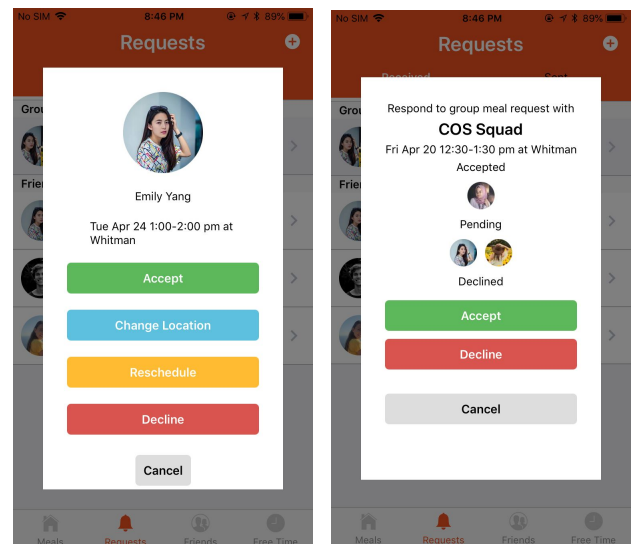
II. Request by Time

Select a time that you’re free. You can see which friends are also free at that time and make a meal request.

III. Responding to received requests

When you receive a meal request from a friend, you can choose to accept it, decline it, or reschedule for a different time.

When you receive a meal request from a group, you can choose to accept or decline it.

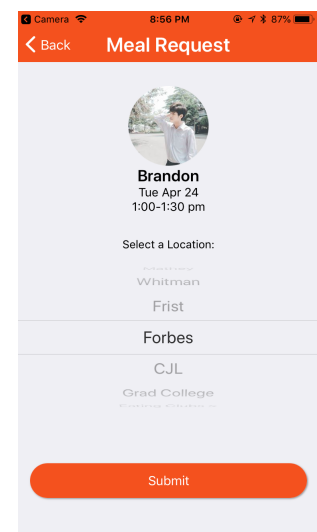


IV. Editing sent requests

After sending a request to a friend, you can reschedule for a different time, change the location of the request, or cancel the request. After sending a request to a group, you can view which members of the group have accepted or declined the meal and finalize it based on their responses. You can also reschedule for a different time or cancel the request.

V. Locations

When making a request for a meal, you can select one of many different dining locations on campus, including the dining halls (Forbes, Mathey, Rocky, Whitman, Wilcox, Wu), the eating clubs (Cannon, Cap, Charter, Cloister, Colonial, Cottage, Ivy, Quad, Terrace, TI, Tower), Frist Campus Center, CJL, the Grad College, or you can enter a custom location.



Developer Guide

Backend

MealUp uses Firebase's [Cloud Firestore](#) to store a large amount of user data. Firebase is a NoSQL cloud database for storing and querying data in a hierarchical structure and syncing with the app through realtime listeners. We store each user by Facebook ID number, and for each user we maintain fields for the user's name, calendar permissions, token (for push notifications), and collections for the user's free times, friends, groups, meals, sent requests, received requests, sent group requests, and received group requests.

Notable Functions:

I. Request by Friend:

We query the user's freetime arrays and query the selected friend's freetime arrays and iterate through the times over the next week, and marking the times when both users are free.

II. Request by Time:

When a user changes his or her freetime arrays, a firebase cloud function is triggered which updates the user's friends' FreeFriends arrays with whether the user is free at that time. That way when a user wants to request a meal by time, we can query the user's FreeFriends arrays and get the list of friends who are free at that time.

III. Rescheduling:

If a user is rescheduling a meal or request, we pass the ID of the meal or request being rescheduled as a prop through the request process to the final request screen, and when the user submits the reschedule request we delete the original meal or request (by ID) from the database for all users taking part in the meal or request.

Database Structure

Users (userID):

- Name
- Token (for push notifications)
- Calendar (permissions, identifying info)
- FreeFriends
 - For each day:
 - For each time:
 - For each friend: friend ID, true if free
- Freetime
 - For each day:
 - For each time:
 - 0 for not available
 - 1 for available
 - 2 for scheduled meal
- Friends (friend userID)
 - For each friend:
 - canViewFriend (true or false)
 - canViewMe (true or false)
 - Name
 - numOfMeals
- Groups (group ID)
 - groupName
 - members
 - For each friend: friend ID, name
- Sent Requests (request ID)
 - DateTime
 - FriendID
 - FriendName
 - Length (duration of meal)
 - Location
 - TimeString
- Received Requests (request ID)
 - same structure as sent requests
- Meals (meal ID)
 - same structure as sent requests
- Sent Group Requests (request ID)
 - same structure as sent requests plus:
 - isGroup (true)
 - members
 - For each friend: friend ID, name, accepted/declined
 - initiator
 - groupName
- Received Group Requests (request ID)
 - Same structure as sent group requests

Frontend

The frontend of MealUp is built with [React Native](#), a JavaScript framework developed by Facebook that allows developers to build native mobile apps in iOS and Android. A simulator and/or mobile device are necessary to run the application. In our experience, the [Expo](#) toolchain & client app served this purpose well.

Frontend Structure & Navigation

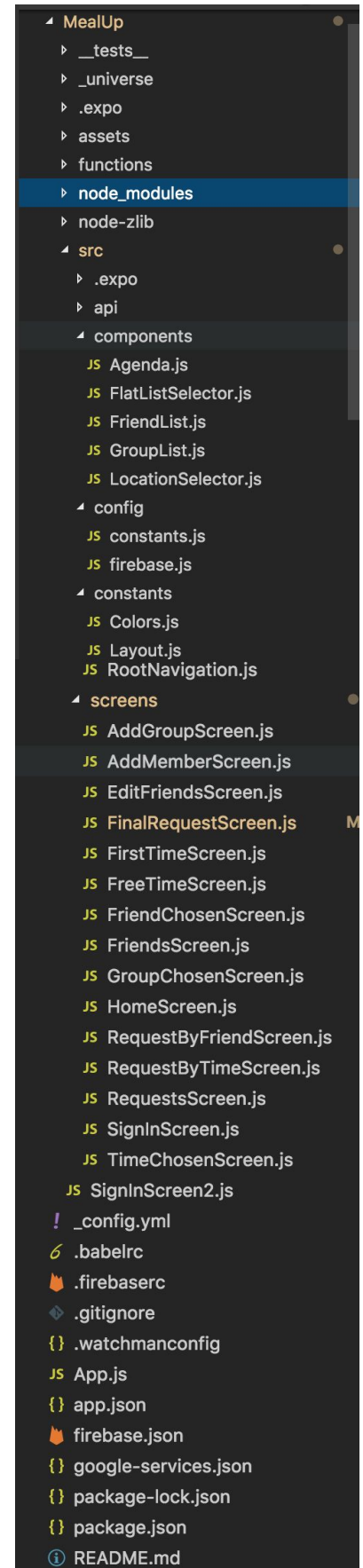
The screens of the app between which navigation occurs are found in the folder `'/src/screens'` within the main directory. This navigation is controlled by `'/src/navigation/MainTabNavigator.js'` and `'/src/navigation/RootNavigation.js'`. To make changes to navigation dealing with a user's first log-in, modify `RootNavigation.js`. Otherwise, modify `MainTabNavigator.js`, which contains MealUp's main tab navigator (Meals, Requests, Friends, Free Time) and four stack navigators, one for each tab.

Of special interest may be the flows for making meal requests based on the `RequestsStack` in `MainTabNavigator`. Users can initiate requests from `HomeScreen.js`, `RequestsScreen.js`, or `FriendsScreen.js`.

- **Starting from HomeScreen and RequestsScreen:** The user selects either 'Request by Friend' or 'Request by Time'.
 - `RequestByFriendScreen`: select a friend, navigates to `FriendChosenScreen` or `GroupChosenScreen`.
 - `RequestByTimeScreen`: select a time, navigates to `TimeChosenScreen`.
- **Starting from FriendScreen:** When a friend is chosen, it automatically navigates to `FriendChosenScreen`.
- When a time is selected in `FriendChosenScreen/GroupChosenScreen` OR a friend is selected in `TimeChosenScreen`, the user is navigated to `FinalRequestScreen` for confirmation and location selection before submitting the request.

Information on screens, in alphabetical order (`=>*` denotes initial screen of a tab):

- **AddGroupScreen.js:** Allows users to create a new group. Renders `FriendList` (import from `FriendList.js`) and optional text box for group name. Reached from the `GroupList` (import from `GroupList.js`) under the Groups tab of `FriendScreen`.
- **AddMemberScreen.js:** Renders `FriendList` (imported from `FriendList.js`) and allows users to add members to an existing group. Can only be navigated to from the `GroupList` (imported from `GroupList.js`) under the Groups tab of `FriendScreen`.
- **EditFriendsScreen.js:** Renders `FriendList` (imported from `FriendList.js`) with the edit option activated and allows users to restrict view permissions for certain friends. Can only be navigated to from `FriendsScreen`.
- **FinalRequestScreen.js:** Renders information for a meal request for confirmation before it is sent and also renders a `Picker` component (imported from `react-native` module) to allow the requester to select a location. When the request is sent, a new document is written for all users involved in the meals in either their Sent Requests, Sent Group Requests, Received Requests, or Received Group Requests. Can only be navigated to from `FriendChosenScreen`, `GroupChosenScreen`, and `TimeChosenScreen`.
- **FirstTimeScreen.js:** Renders onboarding swiper that appears on a user's first log-in. Can only be navigated to once per user from `SignInScreen`, and only navigates to `FreeTimeScreen`.



- ***FreeTimeScreen.js:** Fourth tab from the left of the tab navigator. Imports and uses FlatListSelector from FlatListSelector.js to properly render & color time blocks, meal categories (by time of day), and dates for the upcoming week, based on the current date and time.
- **FriendChosenScreen.js:** Navigated to as part of the request flow after an individual friend is chosen from RequestByFriendScreen. Matches and renders times of the week (both 30-min and 1-hr intervals) when the user and the chosen friend are both available. If there are no matching times, a message about the friend's unavailability is displayed.
- ***FriendsScreen.js:** Third tab from the left of the tab navigator. Uses the Facebook Graph API to fetch the user's Facebook friends who are also on the app and displays the friends sorted in alphabetical order. Refreshes every time screen is rendered.
- **GroupChosenScreen.js:** Navigated to as part of the request flow after a group is chosen from RequestByFriendScreen. Matches and renders times of the week (both 30-min and 1-hr intervals) when all members of the group are both available. If there are no matching times, displays the times when all but one member of the group is available.
- ***HomeScreen.js:** First tab from the left of the tab navigator, and the homepage that a user is brought to on any visit besides his/her initial log-in. Imports and uses Agenda from Agenda.js. Modals for scheduling a meal if no meals scheduled on a day and for canceling or rescheduling a scheduled meal.
- **RequestByFriendScreen.js:** Renders FriendList and GroupList (imported from FriendList.js and GroupList.js) so user can click to select a friend to request a meal with the friend; user will be navigated to FriendChosenScreen, to which the friend selected is passed as a prop.
- **RequestByTimeScreen.js:** Renders the user's freetimes during which at least one of the user's friends is also free. User can select a time to go to TimeChosenScreen to see which friends are available at that time.
- ***RequestsScreen.js:** Second tab, renders list of sent and received requests and button for making requests.
- **SignInScreen.js:** Obtains user permissions for push notifications, authenticates user using Expo's Facebook authentication library, sets up listening for in-app notifications, initializes user's entry in the database with basic structure and data, including Friends collection, Freetime arrays
- **TimeChosenScreen.js:** Navigated to as part of the request flow after an available time is chosen from RequestByTimeScreen. Fetches and displays the user's friends who are also free at the selected time.

Information on frontend components:

- **Agenda.js:** Defines how the schedule of meals on HomeScreen.js is rendered. Refreshes on screen render.
- **FlatListSelector.js:** Responsible for properly displaying free, busy, and blocked times on FreeTimeScreen by fetching a user's Freetime from Firebase.
- **FriendList.js:** Defines how a user's list of Facebook friends under the FriendsScreen Friends tab is rendered. Includes functions to set user's view permissions for certain friends.
- **GroupList.js:** Defines how a user's list of groups of friends under the FriendScreen Groups tab is rendered. Includes functions that allow users to leave, rename, & add members to a group that appear on long-pressing.

Module Dependencies

- | | |
|--------------------------|--------------------------------------|
| • @expo/vector-icons | • react-native-onboarding-swiper |
| • expo | • react-native-scrollable-tab-view |
| • react | • react-native-swiper |
| • react-native | • react-native-vector-icons/Ionicons |
| • react-native-elements | • react-navigation |
| • react-native-calendars | • react-navigation-header-buttons |
| • react-native-navbar | • rn-prompt |

Acknowledgements

We'd like to thank Jérémie Lumbroso for advising this project throughout Spring 2018.