

PhotoMap

Test iOS Application

Application Description

PhotoMap Mobile is an application which allows to make and organize photos. Each new photo should be placed on a map where a photo was taken.

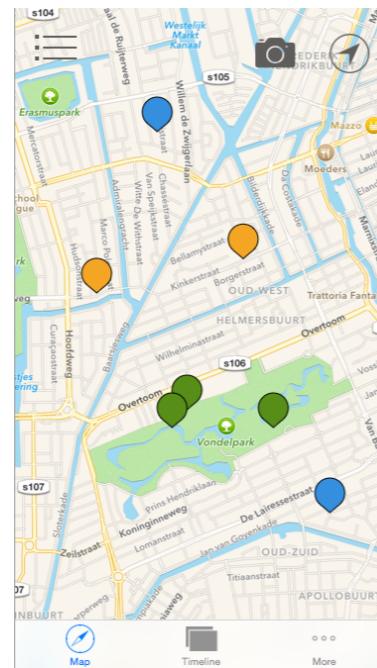
Users should have possibility to add a description to a photo, assign a photo to a category, find needed photos by hashtags.

All user-generated content should be saved on a backend side. The app uses firebase.com as a backend solution.

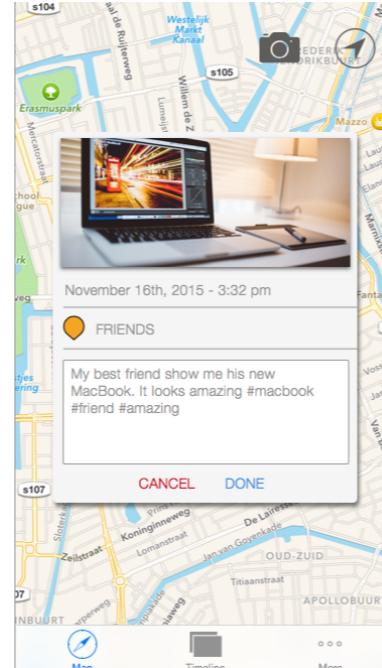
Please create new application on firebase.com and integrate Firebase SDK into your project as described here (<https://www.firebaseio.com/docs/ios/guide/setup.html>)

Please create a repository on Bitbucket (<https://bitbucket.org/>) and invite your project lead to the repository.

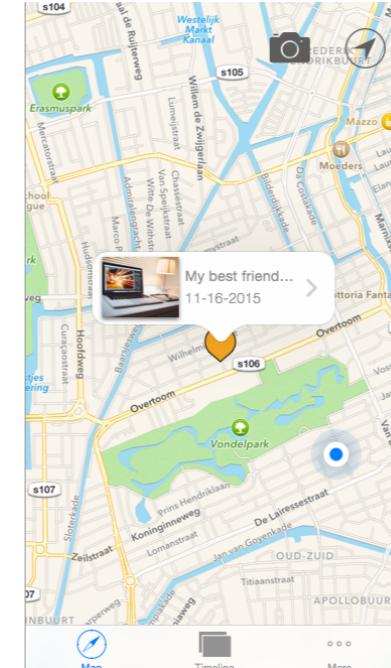
PhotoMap Application - Main Screens



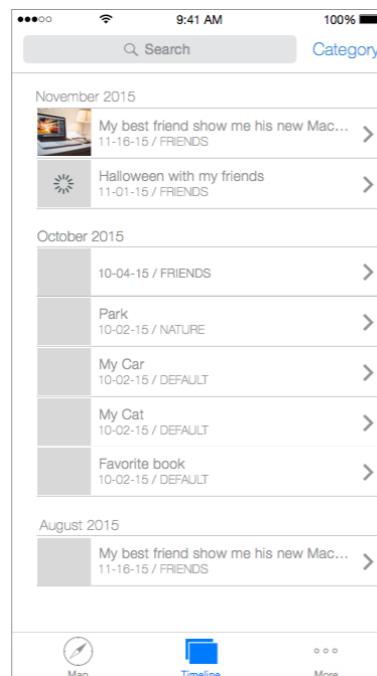
Screen 1.1 - Map



Screen 1.2 - New Photo Popup



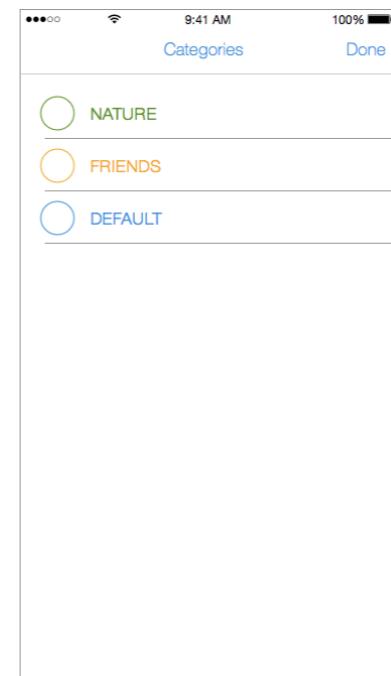
Screen 1.3 - Photo Marker Popover



Screen 2.1 - Timeline

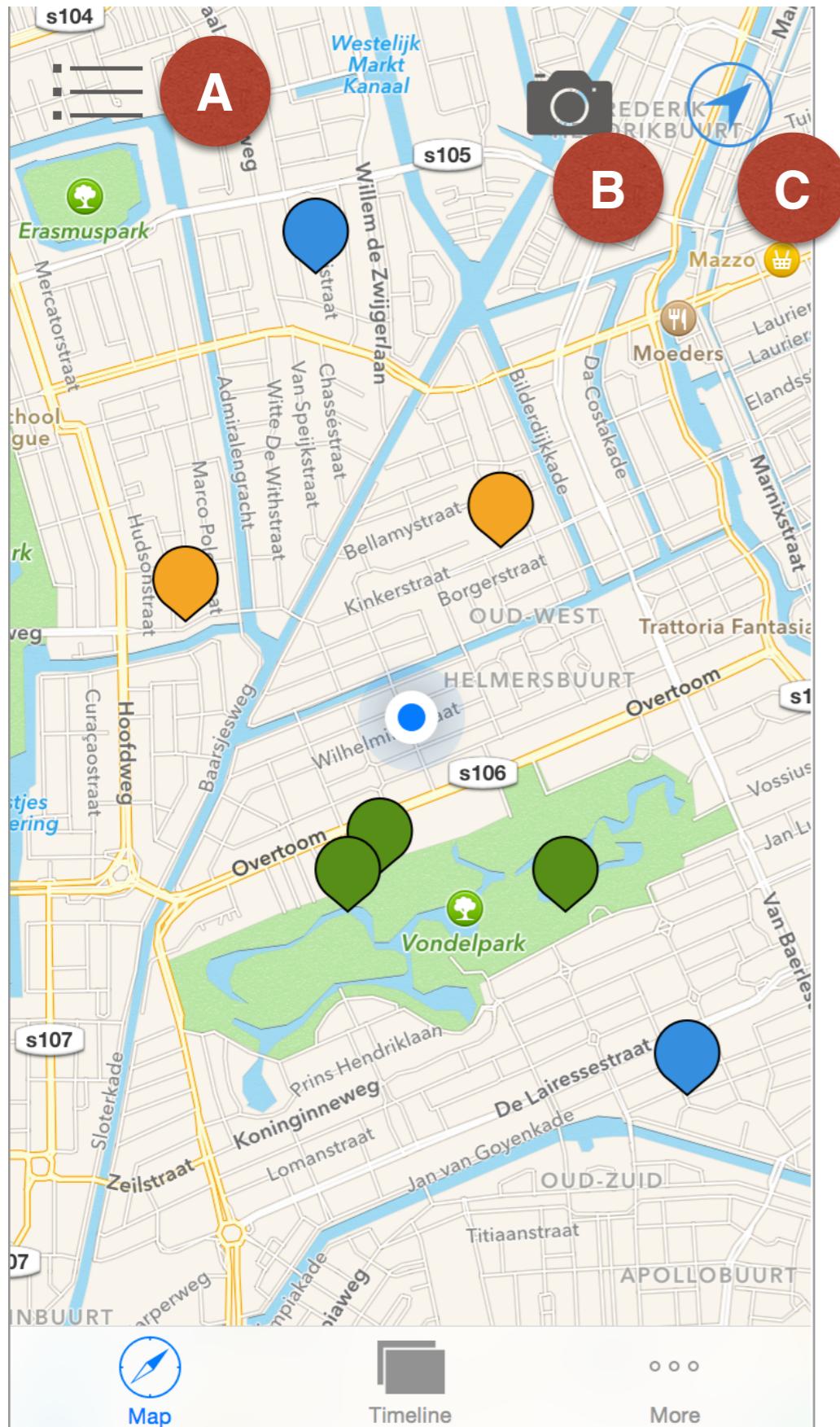


Screen 2.1 - Full Photo View



Screen A.1 - Categories

Screen 1.1 - Map



Screen 1.1 shows Map View and Photos which were made by the user in PhotoMap application. Photos are displayed as colourful markers on the map. Markers can have different colors depending on a category of the image. There are 3 available photo categories:

- Friends -
- Nature -
- Default -

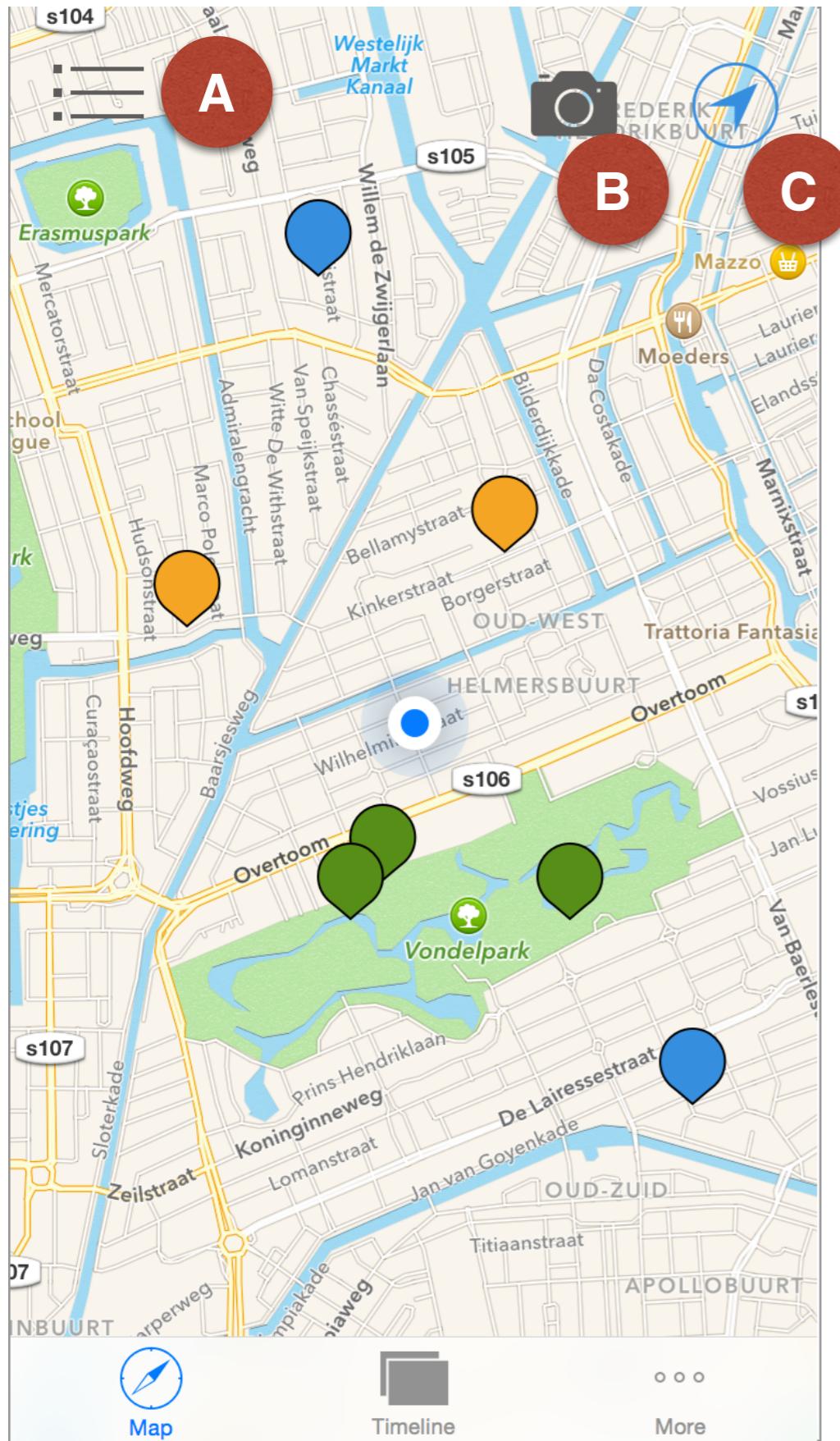
By default, all categories are displayed on the map, but the user can choose which categories should be shown. User can filter categories by Category Button (A). Category Button opens modal Screen A.1, where the user can select needed categories. For example, if the user unselects Friends category, only Nature and Default photos should presented on the map.

Map Navigation

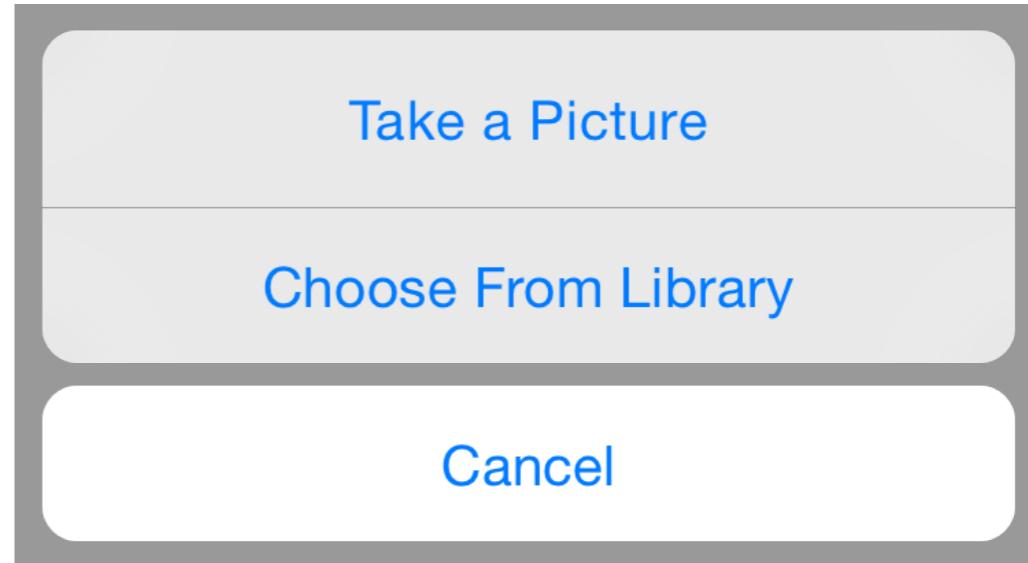
Map View provides 2 Navigation Modes: Follow and Discover. Modes can be switched by Navigation Button (C). By default, Follow mode is enabled (Navigation Button color is #368edf). Follow mode always centers the map on current user's position. If the user moves, the map follows the user, so the user is always in center.

Discover mode (Navigation Button color is #5F5C5C) is enabled if the user moves the map around (e.g the user drags map by finger). During this mode the map doesn't follow the user.

Screen 1.1 - Map



The user can make a photo by pressing Photo Button (B) or by long-press action over Map. In both cases the app should display an ActionSheet as shown below:

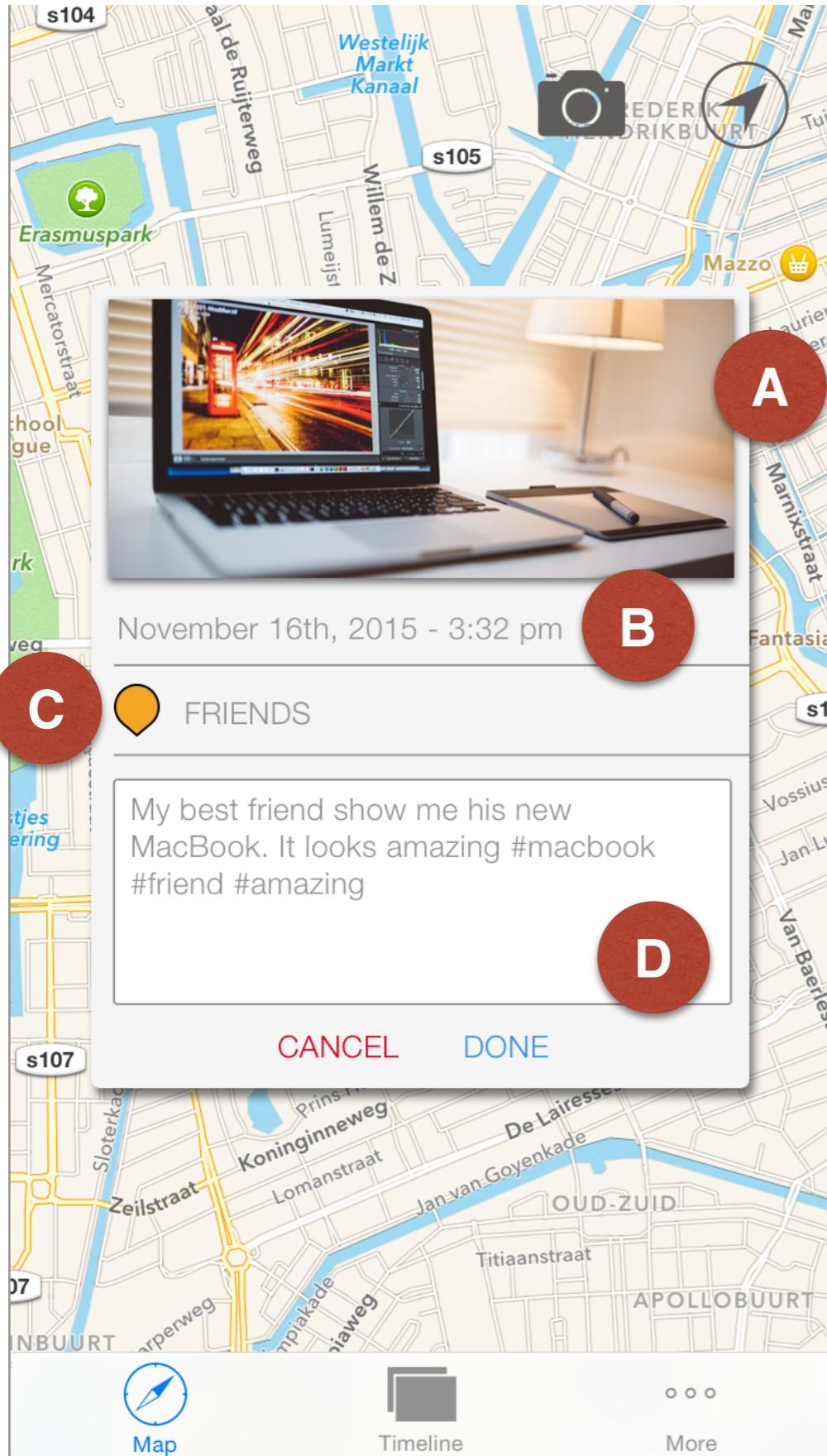


- Take a Picture button should open Phone Camera
- Choose From Library should open Photo Gallery

If the photo is created using Photo Button (B), the photo marker should be placed in current user's location

If the photo is created using long press, the new photo's coordinates are defined by the place, where the user touched the map.

Screen 1.2 - Map Photo Popup

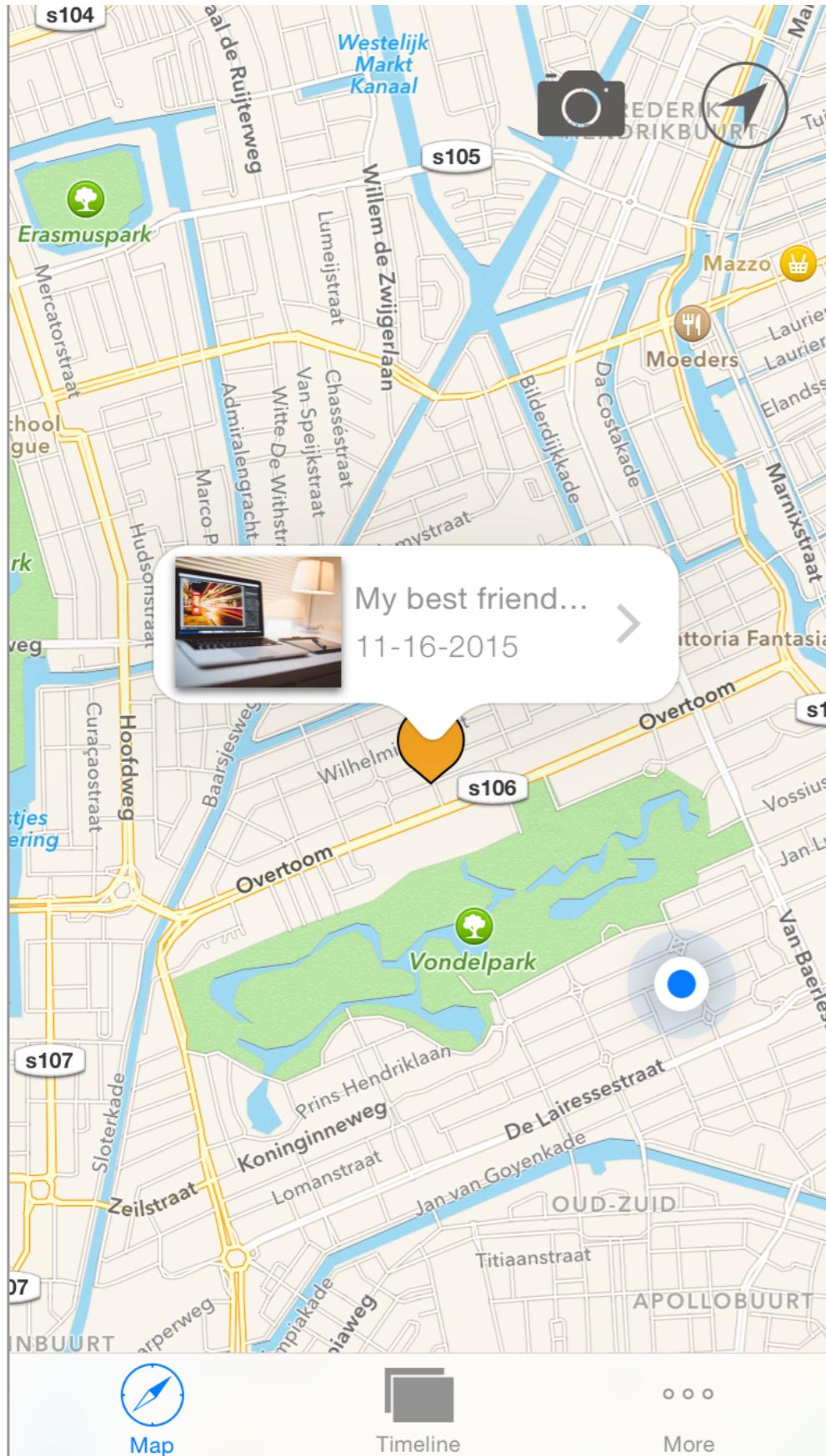


When a photo is created, Map Photo Popup should appear. This popup is used to add a description to a newly created photo and assign the photo to a category. Popup consists of the following items:

- (A) A preview of created photo. Should be clickable. If the user clicks preview, full photo view should be opened (Screen 2.1)
- (B) Date and Time when photo was created (implement the same format as it's shown in the picture)
- (C) A category the photo is assigned to. User should be able to change a category. When the user clicks this field, a modal picker view should appear. Picker view contains all available categories (Nature, Friends, Default).
- (D) Editable text field. The user should be able to add a description to the photo. Optional field.

If the user clicks Done button, new instance of the photo should be created and uploaded to Firebase database.

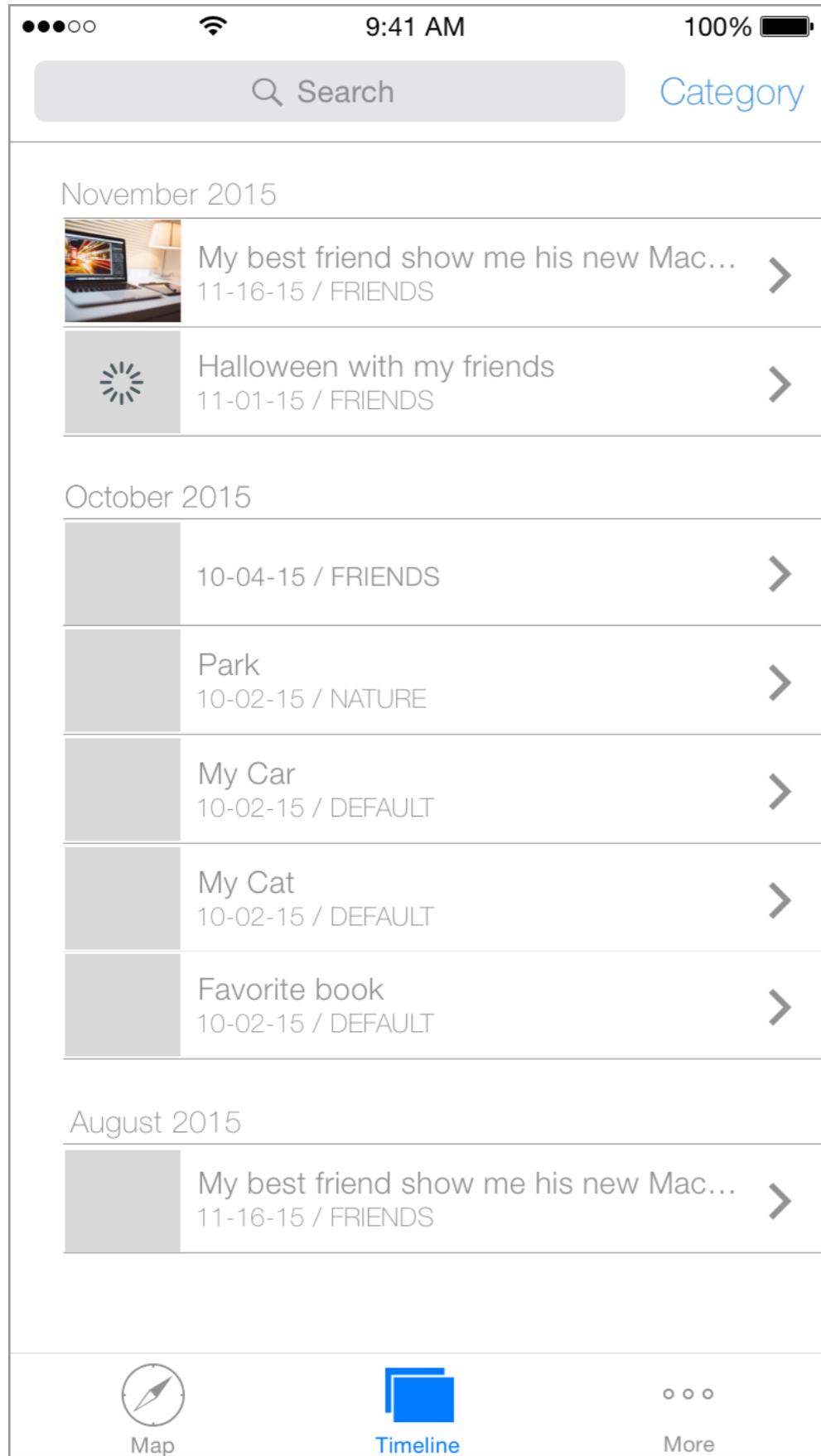
Screen 1.2 - Map Photo Popup



All Markers on the map are clickable. If the user clicks a marker, a popover should appear above the selected marker as shown in the picture. The map should be centered on selected marker.

If the user clicks the popover, Map Photo Popup (Screen 1.2).

Screen 2.1 - Timeline



Timeline screen shows the list of all the photos created by current user. The list is sorted by historical order: the most newest items at the top. The list is divided into months as shown in the picture.

Each item in the list (cell) shows the picture, description of the image (if available) at the first line, date and category title as the second line. If description is not available, only date/category should be available

Users should be able to filter categories using Category button at the top right corner (Category button opens Screen A.1).

Users should be able to Search through the list by photo hashtags (hashtags can be added as a part of description)

If the user selects a cell, Full Photo View should be opened (Screen 2.2)

Screen 2.2 - Full Photo View

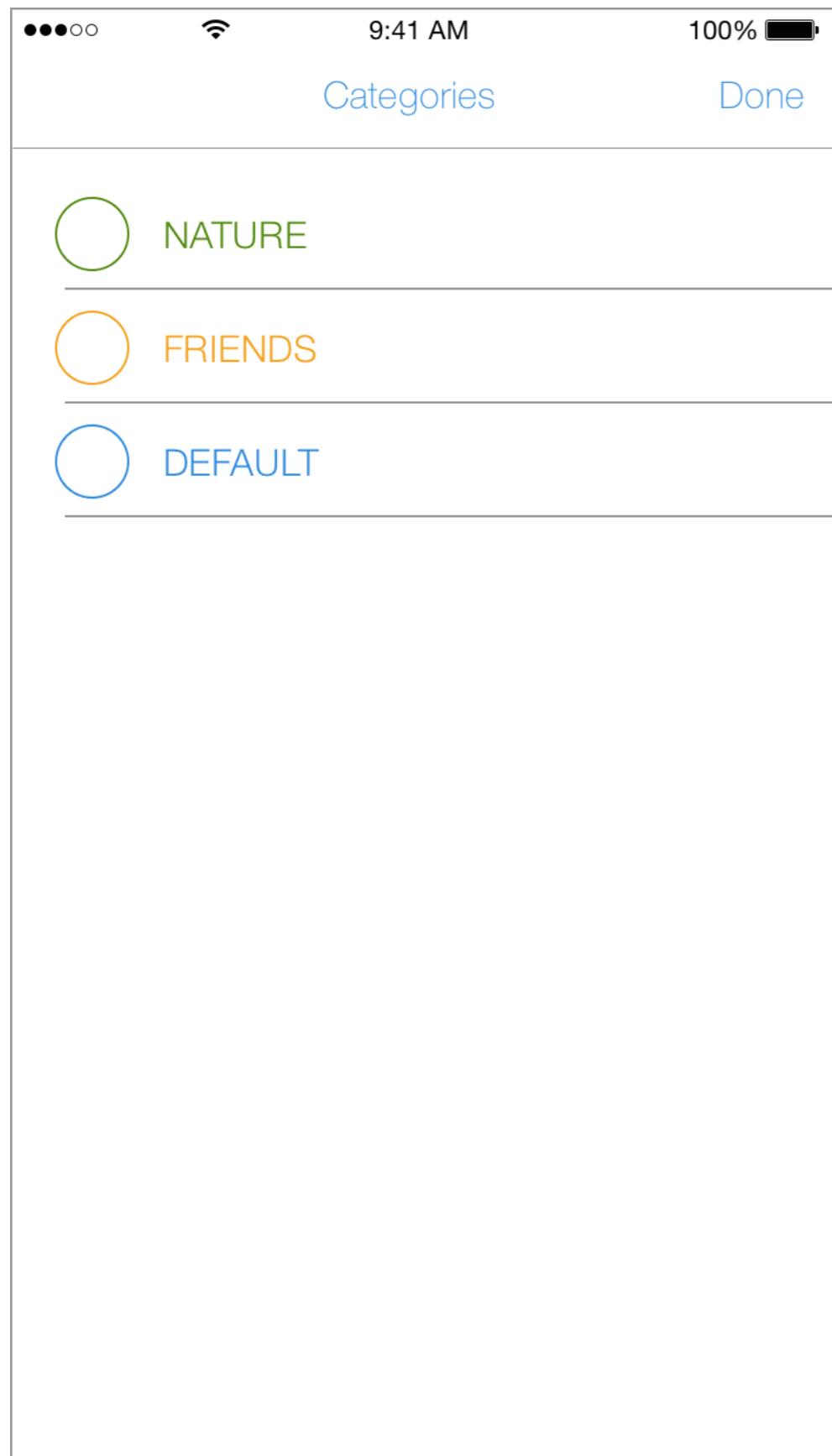


Full Photo View shows fully-visible selected photo. The photo should be zoomable (by double-tap and pinch-gesture)

The screen shows Header (A) and Footer (B). Header and Footer can be hidden by single tap on the photo. They appear if the user taps on the photo again.

Footer shows description of the photo (if available) and data when the picture was created

Screen A.1 - Categories



Categories Screen allows to filter photo categories.

A category can be selected/unselected by checkbox button (rounded colorful button). By default, all categories are enabled (checked)

Example of selected categories

- | | | |
|--|---------|----------------|
|  | NATURE | Color: #578E18 |
|  | FRIENDS | Color: #F4A523 |
|  | DEFAULT | Color: #368EDF |

Example of unselected categories

- | | |
|---|---------|
|  | NATURE |
|  | FRIENDS |
|  | DEFAULT |

11 Additional Information

- Implement Login and SignUp Screens using default iOS controls.
- iPhone only support
- iOS 11+ support
- Don't use third-party frameworks to download images. Please implement your own image downloader
- Try to implement as many UI items in XIBs as possible
- You can use Storyboard