

Jennifer Haigler(jrh7qp), Caroline Darch(cmd3az)

Ccqpr04FGGK5 (“pink lady”)

## Pensieve

The Pensieve app is designed as a memory tracking device. Every time an important event comes up and you would like to keep track of it, create a new memory. You’ll be able to store the location, date, time and photos in discrete memory points. Revisit a memory at any time to review the progress of any event. Look through photos of the day, and be able to pin point exactly where and when they took place. Keep your memories organized and never run the risk of forgetting “where were we when we took that photo?” or “was that Christmas of last year or the year before?” Pensieve allows you to forget the minutia without letting the details be lost forever.

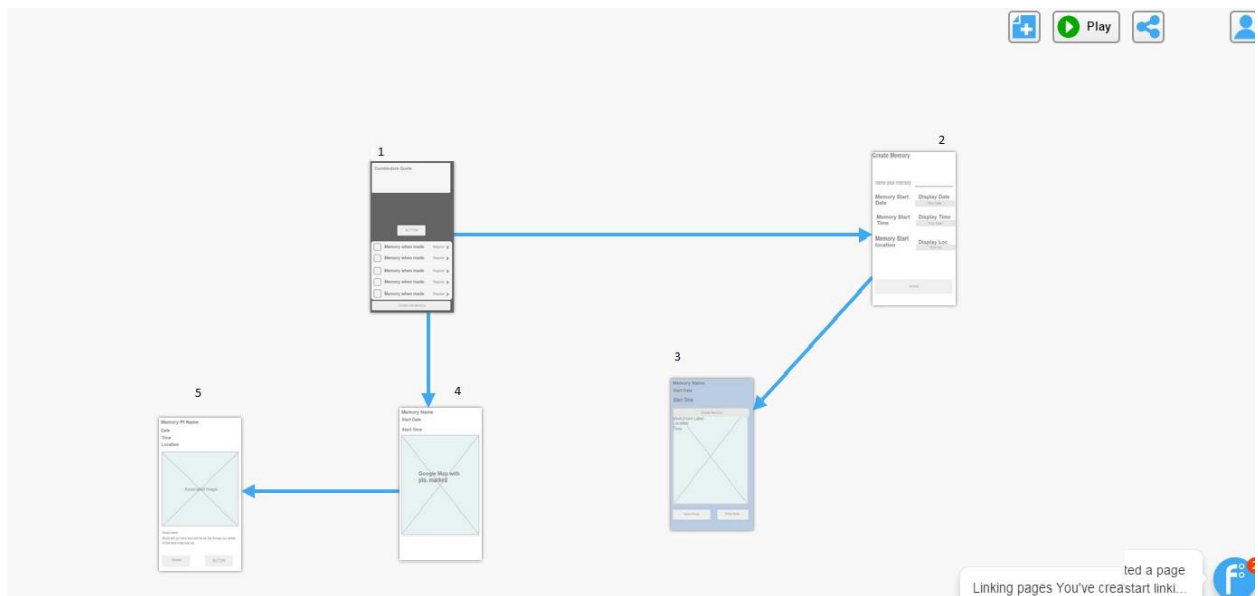
- Five Scenes
- GPS
- Camera sensor
- Use of existing Photo Gallery
- Portrait and Landscape integration
- Google Places
- SQLite storage through CoreData
- Internal storage for photos

This app implements a SQLite database which is used to store all the details associated with each memory and each memory point. The main screen provides an updated table view of all previous memories made, and by clicking on an item of the view, the user is taken to a table view of memory points corresponding to the memory. Within a memory, the user has the option to look at previous memory points or create new memory points. Pensieve utilizes the camera hardware on a device to take pictures for each memory. Photos can also be added to a memory point from the existing photo gallery of the device. For location data, the application utilizes Apple Maps to associate nearby locations by name and address to memories and memory points.

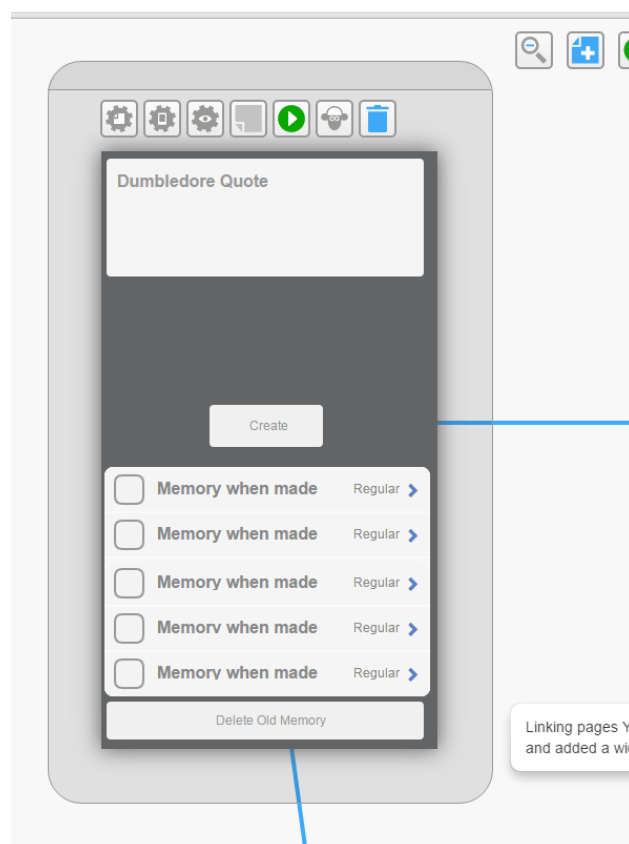
No special information is needed to run the app, just your apple device. Upon opening the application you will be brought to a page that allows you to create a new memory or view old memories if they have been created. If you choose to make a new memory, you will be prompted to name it and give a date, time, and location you would like associated with it. From there you will be led to a screen that allows you to go to a create memory points screen or to view memory points already associated with your memory. If you choose to create a memory point, you will again be prompted to give a date, time and location associated with your memory point. You will also be able to take a picture to associate with the point. When you hit “save” a new memory point will be created, and then you will be directed back to the prior screen. Your new memory

point should now show up on the list of memory points. Memory points can be viewed by clicking on this list. If you initially choose to view an old memory you will be directed to the page with a list of the memory points for the selected memory. You can then add points to a memory or view the memory points saved to it.

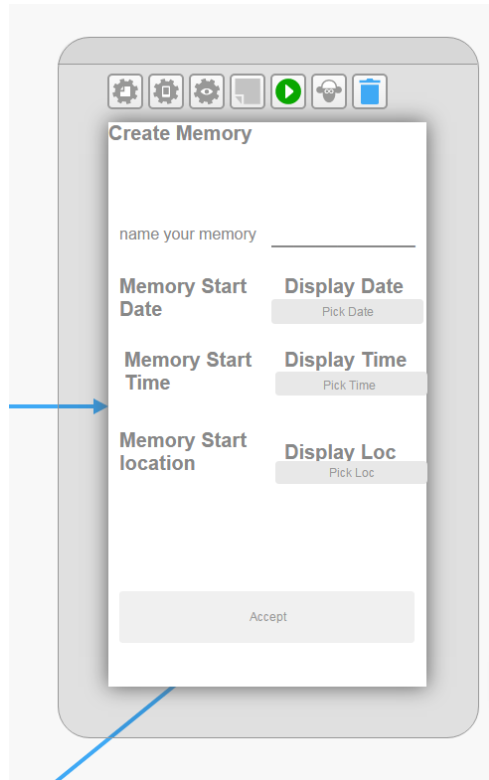
One of the biggest problems we encountered while coding this app was the update of xcode and iOS. We found it difficult to find the information we needed to implement certain features, and when there was information, it was for an older version of iOS or it was in objective C or it made reference to the now non-existent .h files. As a result, we learned most of what we learned from tutorials that had been designed by apple or apple developers specifically for the new xcode. It was slow going and almost all of the layout design we implemented (the scroll views for example and the stack views we used) were done on a basis of trial and error. As a result, we now know quite a bit more about how to make constraints for views. We became very good at segues and navigation by the end of it as we kept running into problems with navigating back. We got into the habit of using segues when we needed views on previous screens to become completely updated instead of just unwinding. The navigation bar buttons became our “uh-oh” buttons for the most part that would undo navigation if a mistake was made. There were exceptions to this, like going from the memory point list in a memory back to the beginning screen using a navigation bar button and adding points from the start screen, but for the most part we made up our own set of standards that gave the app its own internal logic without departing from apple’s conventions. We redesigned the way we did the table this time because we worried about overlap in memory points made at the same time stamp on the same date. This was something we discovered was a problem during android, so each mem point is now identifiable by a number that is saved in the table. This allows us to avoid unreachable entries because all of their table information is identical to another memory points.



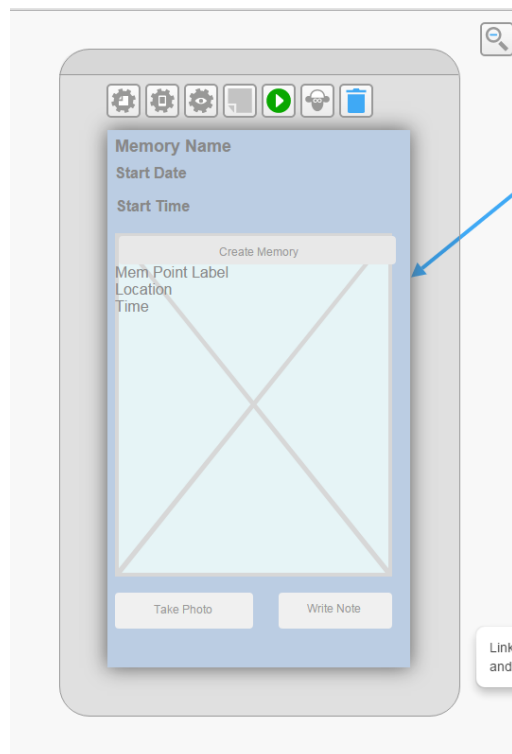
Wire Frame Map



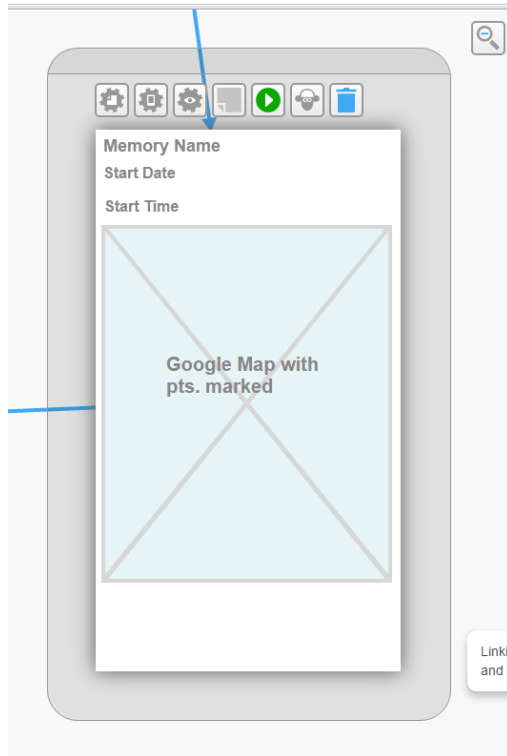
Screen 1



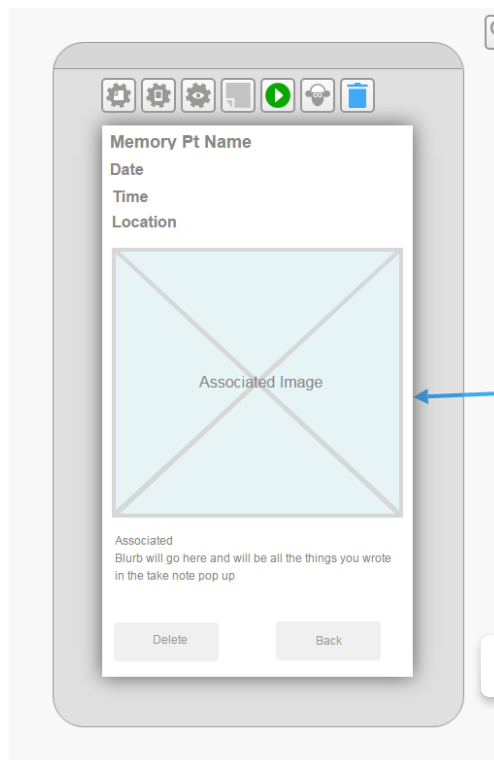
Screen 2



Screen 3



Screen 4



Screen 5