Time Capsule Final Report

Team: Vibhu Yadav, Richard Palomino, Cristian Caraballo

Introduction

When we came up with the idea of a Dartmouth time capsule we had to sit down and think about the service it would provide to the Dartmouth community. The amazing thing about Dartmouth is that everyone, of course, has their own unique experience during the four years they stay here. These experiences, for the most part, remain personal and aren’t recorded nor is there a place for these experiences to be documented. When designing the TimeCapsule application we wanted to make it accessible for everyone to chime in and give their 2 cents in regards to their own, unique Dartmouth Experience---provide a medium in which every individual’s experience can be shared on this campus.

So the idea is that once the application is launched, Google Map is also launched and a number of location pointers are laid out in the map already. The locations that we chose are generally considered “hotspots” on campus like The Green, Baker, The Hop, etc. The user can click on a location marker and then choose to whether add a new capsule or view the capsule entries of that particular location on campus. TimeCapsule is simple and concise purposely. We believe that time capsules should be used as preservatives of the past and present We believe that emotion, thoughts, and memories are the content that should be preserved and shared for the future. Our TimeCapsule does not strive to be: a social network, personal messenger, or personal journal. Our goal was to make TimeCapsule a community archive of the Dartmouth Experience.
Architectural Design

The design of the application is broken into four activities and one service.

The **MainActivity** class is populated by the Google MapFragment which is bundled with a GoogleMap class. The Google MapView is populated by markers with custom icons, and info windows that display the title and short description of the Landmark. Upon clicking on a marker, the InfoWindow is opened and can be clicked to reach the Notification Activity.

The **NotificationActivity** class presents a Landmark’s information. The “Landmark” of the TimeCapsule application are points of interest that are specific to the Dartmouth campus. The NotificationActivity thus shows the name, stock image, and more detailed information of that Landmark. The second half of the NotificationActivity is dedicated to listing the Capsules that are stored in the database for that location. The ListView is backed by a **CapsuleListAdapter** that pulls each Capsule’s title for display from the **AdapterBinding** object for that Capsule. The AdapterBinding object also holds the database’s _id column value for that Capsule to open the CapsuleViewActivity and supply the intent with id so that the next Activity can query the database for the specific capsule.

The **CapsuleActivity** class is the builder of a **CapsuleEntry** object. The CapsuleActivity allows users to input all of the CapsuleEntry fields via EditText’s that match where **CapsuleViewActivity** places the finished capsule information. The FrameLayout that displays a camera icon can be clicked to replace that camera icon with an actual JPEG that is saved to the CapsuleEntry in the database. The **CapsuleViewActivity** essentially matches the layout of the CapsuleActivity but pulls the CapsuleEntry that has the intent-supplied id, and then displays the information and picture.

The **LocationService** runs as soon as the MainActivity is opened and until the Notification is cancelled by a swipe from the user. The LocationService uses a Google LocationClient to receive periodic updates that will return the user’s current location. The latitude and longitude are extracted from the location update and compared to the Landmarks’ locations to determine which is closest. A Notification is then generated and sent to the NotificationManager to be displayed, showing the closest location. Upon clicking this Notification, the NotificationActivity is opened with the same closest location’s information.
TimeCapsule Team Member Roles

For much of the project, we did not split the application into distinct parts to work on. To begin with, we all sat down and decided what the application should do and had a very loose idea of what the UI would look like. We also decided that for simplicity’s sake, we would focus on specific places to store Capsules before allowing for free placement of Capsules on the map. These would also become permanent default locations that could be used instead of free placement. We then decided that each Capsule would then only contain text and a photo to begin with. We then started to work on the application in successive parts, in which we each would code a piece of.

We initially broke up the work into overall components, Richard wrote very basic skeletons for the activities and service, while Cristian wrote the CapsuleEntry object and Vibhu wrote the SQLiteOpenHelper class for the local database.

We started first with the NotificationActivity which shows which Landmark you are closest to if entered from the Notification or the specific Landmark that you clicked on from the GoogleMap within the MainActivity. We all agreed on a basic layout of the Notification and the NotificationActivity. Richard then created the actual xml of the layout as well as starting with the information/photo gather for Sudikoff Lab, which became the first Landmark. Richard also created the skeleton for the BigPictureStyle Notification that shows the current closest location. After all looked well, Richard created the MainActivity’s layout as well as coded the MainActivity class and a file parser to create markers for the included stock Landmarks that can be opened by clicking the markers’ InfoWindow.

We then began to add in code for the Capsules. Vibhu worked on the database methods which would be used to insert and retrieve capsules as well as store landmark metadata. Cristian went on to design the layouts and basic callbacks to store everything minus the pictures from the Capsule for the CapsuleActivity. Richard and Vibhu also finished the LocationService’s selection of the current closest location soon after. This was the amount of functionality that the application had at Checkpoint time.

Moving into the final phase of the project, Cristian finalized and redesigned the CapsuleActivity and CapsuleViewActivity, trying to make maximum use of the screen and scale to different sized devices. Richard designed the iconography and graphics of the application. This included the Android Holo-colored capsule images that can be found on the NotificationActivity’s list of capsules and the capsule cutout, signifying a missing capsule in the list that would be added by the user as. Basic backgrounds were also created for the NotificationActivity and some of these graphics were used for the CapsuleActivity and CapsuleViewActivity. Vibhu added more Landmarks to the list of default Capsule Locations.

At this point all activity flow was finished, and the LocationService was finalized. Richard wrote the final implementation of the NotificationActivity, showing Capsule Count and the CapsuleListAdapter-backed listview with holo-colored capsules. Richard also wrote the quick-capture camera functionality that shows the capsule photo and allows you to take a pre-sized image within the application, as well as the saving and retrieving functionality to and from the local database. Cristian fixed a few bugs that occurred along the way and made sure that the current application is completely stable. Richard concurrently began writing the Application-side of the GoogleAppEngine Cloud IO, creating the POST method and JSONContract used by the two sides. Vibhu began the Google AppEngine Datastore implementation to store Capsules in the cloud, but that is not finished in the the final version of the project.
TimeCapsule Running Instructions

Once the apk has been downloaded and the TimeCapsule application is launched---Google Maps will be launched as well and you will see a map of Dartmouth campus with multiple location markers. The user can click on any of these location markers **and then click on the name label** to open the Notification Activity. In this activity, the user will see a picture of the location, information about the location, along with any capsule entries that have been committed to that particular place. There will also be an **Add Capsule** button that the user can click on if he wants to record a memory at that location.

If the **Add Capsule** button is pressed---this is the New Capsule screen. Here you can record a picture, username, event name, and thoughts. To input an event name the user must click on the little pencil on the top right of the screen. To take a picture the user must click on the picture of the camera to launch the camera. And the username/thoughts are simply editText views where the user can just click on them.

If the user chooses to view other capsule entries---this is the Capsule View screen. This view is very similar to the New Capsule screen with the only difference being the buttons and of course, you can’t edit the capsules from this view, or any view for that matter. This view has a like and dislike button. This is designed for others who read the capsule entries and can like or dislike someone else’s entry.

This Notification screen can also be launched through the pull down view of Androids. The application is always keeping track of the nearest TimeCapsule location to the user. So in the pull down window you will see a TimeCapsule tab with the nearest location and if you click on it the Notification screen will be launched and you can either view other entries or add a capsule entry of your own.
What did we learn from this project?

We learned android can be so much fun. We got to learn a whole lot of things that make android so amazing. We struggled at places but whenever we resolved it, we felt the excitement and the joy of progress. We also learnt how important team-work is and we realized being more organised would have resulted in a much better app in the due time. We also learnt the importance of time management in group projects and communication between the members.

On the technical side we learnt more stuff. We learnt lots of SQLite3 database and its functionality in Android. We learnt how to use the location services and map API for various uses in the application. We also got to know a lot of UI design and some photoshop. We also learned a whole lot of cloud and google app engine. Given the time, we could not complete it but it was really fun.

We learnt how every android phone is different from others and fitting the UI and other stuff is as important as coding the application. We struggled to setup the UI correctly on different android devices but overall it was all fun. We learnt to keep things simple and clean and yet not destroying the functionality of the application.

Overall the project was an awesome learning experience. Not only did we learn the android programming but also how to efficiently work in a group and learn from our mistakes. We greatly thank Prof. Andrew T. Campbell for this opportunity.

Conclusion

TimeCapsule was a fun experience. The App does what it intends to do but we hoped we could get it to store to the cloud and students around the campus would have begin sharing their experiences here at Dartmouth. The application is a beautiful way to share your memories and keep it in a capsule with priceless pictures and information. The application will also update locations and inform the user of the same in the notification when travelling around the Dartmouth Campus. We have plans to add more locations. Right now user can use the application store his/her memories on their phone and view later. We will add cloud to the application at earliest for the students to use and hope students make best use of this application. We take this platform to thank Professor Andrew Campbell and Teaching assistants for this opportunity and valuable suggestions, advice and input.